University of Southampton

Faculty of Electronics and Physical Science

Electronics and Computer Science

MITRE Open CTI Contribution to Cyber Situational Awareness

by

Thesis for the degree of PhD Computer Science

August 2024

University of Southampton

Abstract

Faculty of Electronics and Physical Science

Electronics and Computer Science

Doctor of Philosophy

MITRE Open CTI Contribution to Cyber Situational Awareness

by

Christopher John Maidens (20305966)

A cyber-attack is executed through a series of steps to compromise the security of a target's cyber assets. Due to the ever-increasing reliance on computer and network systems to implement critical government and commercial operations, cyber-attacks have become significant threats with potentially severe consequences. Within existing research there is a constant and still outstanding issue around the lack of openly available data to use while testing attack detection algorithms. This is particularly true regarding sources of data describing real attacks in terms of the sequencing (the series of steps) of the Tactics and Techniques employed. These sequences can provide analysts with additional specific information about the behaviour of attackers over and above just a list of the techniques that they use. MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) is a knowledge base that includes descriptions for over 100 significant APTs and the Tactics, Techniques, Tools, and Procedures (TTPs) that they use. This does not, however, include any knowledge about the sequencing of specific attacks. This thesis provides a proposal to address this lack of available attack sequence intelligence and so increase the contribution that it can make to cyber situational awareness. It presents a model that can be used to record data representing a sequence of MITRE ATT&CK TTPs (an ordered set of Tactic and Techniques) observed during attacks. The model also allows the analyst to record relative timings of the steps taken and to associate each step with a kill chain model view of a cyber-attack. The population of this model is exercised using a representative set of open-source attack reports and several example applications are presented.

Table of Contents

Table of Tables

Table of Figures

Table of Figures

Table of Figures

List of Accompanying Materials

Code and data used in this document can be found at (Maidens, 2023)

Research Thesis: Declaration of Authorship

Print name: CHRISTOPHER JOHN MAIDENS

Title of thesis: MITRE Open CTI Contribution to Cyber Situational Awareness

I declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. None of this work has been published before submission

Signature: ..... Date:.....

Acknowledgements

Thanks to Professor Vladimiro Sassone and Dr Leonardo Aniello for their support in this work.

“One can lie on the ground and look up at the almost infinite number of stars in the night sky, but in order to tell stories about those stars they need to be seen as constellations, the invisible lines which can connect them need to be assumed.” - John Berger from Another Way of Telling.

Definitions and Abbreviations

ATT&CK ..... Adversarial Tactics Techniques and Common Knowledge

Blended Attack ..... Where an attacker may deploy multiple approaches and exploits following initial access

CAPEC ..... Common Attack Pattern Enumeration and Classification

CAR ..... Cyber Analytics Repository

CNA ..... CVE Numbering Authority

CPE ..... Common Platform Enumeration

CTI ..... Cyber Threat Intelligence

CTMC ..... Continuous Time Markov Chain

CVE ..... Common Vulnerabilities and Exposures

CVSS ..... Common Vulnerability Scoring System

CWE ..... Common Weakness Enumeration

CyBOK ..... Cyber Security Body Of Knowledge

Cyber Actor ..... Individual or organisation operating in a world of globally connected networks of hardware, software and data

ENISA ..... European Union Agency for Cybersecurity

GAN ..... General Adversarial Network(s)

GEXF ..... Graph Exchange XML Format

IOC ..... Indicator of Compromise

ICS ..... Industrial Control Systems

IDEA ..... Intrusion Detection Extensible Alert

IDMEF ..... Intrusion Detection Message Exchange Format

IDS ..... Intrusion Detection System

JSON ..... Javascript Object Notation

Lateral Movement ..... Phase of APT campaigns where the attackers try to compromise additional systems within the internal environment of the target

LMCKC..... Lockheed Martin Cyber Kill Chain

LPG ..... Loopy Belief Propagation

MANET..... Mobile Adhoc Networks

MISP ..... Malware Information Sharing Platform

NIST ..... National Institute of Standards and Technology

NLP ..... Natural Language Processing

NVD ..... National Vulnerability Database

OSINT..... Open-Source Intelligence

OWL..... Web Ontology Language

Pivoting..... Describes the act of tunnelling traffic through one system to connect to other internal systems

RDF-S ..... Resource Description Framework Schema

REST..... Representational State Transfer

SCAP ..... Security Content Automation Protocol

STIX..... Structured Threat Information Expression

TTP..... Tactics, Techniques/Tools & Procedures

UKC..... Unified Kill Chain

WSN..... Wireless Sensor Networks

Chapter 1 Introduction

1.1 Overview

A cyber-attack is actioned by a malicious actor through a series of steps attempting to compromise the security of a target's cyber assets. There has been an ever-increasing reliance on networked computer systems to implement critical government, commercial and social operations and so cyber-attacks have become (and remain) an important threat to our society with potentially severe consequences.

In considering approaches to detecting cyber-attacks two major categories of approaches can be considered:

- Pattern recognition
- Anomaly detection (Ye et al., 2004)

One area of investigation related to cyber-attack 'pattern recognition' includes the study of attack 'signatures' ("a unique arrangement of information that can be used to identify an attacker's attempt to exploit a known operating system or application vulnerability" (Broadcom, 2023)). Further to this, one 'signature' that could be investigated is the sequence of a particular type of observed event taken by an attacker over time. Depending on the observations being collected and the precise application, there are many types of these sequences that can be observed. Bianco's pyramid of pain provides a useful pen picture to understand some of these general perspectives (Bianco, 2014).

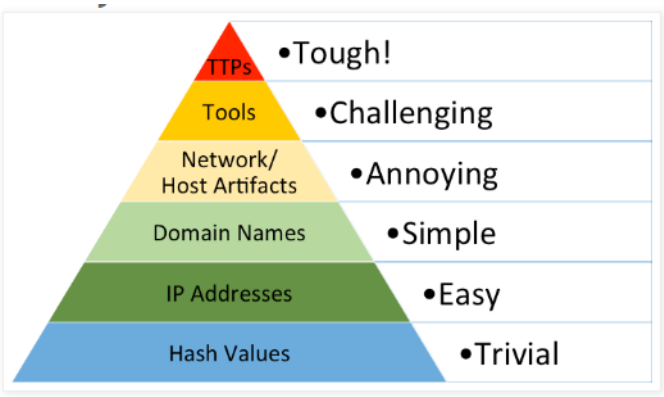


Figure 1 - Bianco's Pyramid of Pain

The lower four levels of the above pyramid are often one-dimensional indicators of compromise (IOCs) and are relatively easy for an attacker to change. The higher levels (representing attacker behaviours) are much more costly for the attacker to change but are also much more difficult to detect and analyse in a dynamic environment.

MITRE ATT&CK provides an important contribution to the study and codification of an attacker's behaviours during an attack. This knowledge base provides several components

- Standardised descriptions of Tactics. Techniques and Tools with supporting notes on observed Procedures (TTPs). These can be used by analysts to create reports that include consistent descriptions of attacks and the related procedures.
- A description of several significant APTs and their use of these TTPs in terms of these standardised descriptions
- A substantial corpus of references to openly available attack reports that are used to support and justify content.

A related MITRE initiative is the Cyber Analytics Repository (CAR). CAR is a knowledge base of analytics developed by MITRE based on the MITRE ATT&CK adversary model. CAR provides a set of pseudo-code of analytics (and some specific tool implementations) that can be used to sense and detect techniques.

Within existing research papers there is a general constant and still outstanding issue around the lack of openly available data to use while testing approaches to understanding cyber-attack behaviours. This is particularly true regarding sources of real attacks described in terms of a sequence of Tactics and Techniques. It is this area that will be investigated in this thesis (see also Proposal Outline).

As mentioned above, the MITRE ATT&CK Knowledge Base includes descriptions for several significant APTs and the TTPs that they use. However, the analytic use of this knowledge base is limited when investigating sequences of suspicious TTPs as it does not include any codification of the sequences of TTPs used by these APTs. The attack descriptions are simply provided as references within the knowledge base. These are written only in natural language with expert analysts in mind and do not standardise how the attack sequences are described. This means that analysts may only investigate specific techniques observed and not the different sequences of these techniques. If for instance two APTs use the same set of techniques, say $\{T_1, T_2, T_3\}$. Then if these are observed in a system, there is currently no additional intelligence that can help the analysts understand characteristic sequences of these techniques used by each APT. Having this additional insight can help analysts target their investigations further.

This work provides a contribution to address this lack of available attack sequence intelligence described as a sequence of MITRE ATT&CK TTPs.

It does this in the following way (see also Chapter 4 below).

- Firstly, an approach is developed to model cyber-attacks as sequences of TTPs. To our knowledge a similar model has not been defined in any research literature. MITRE ATT&CK is not currently used to investigate attack sequences due to this constraint (e.g. Summary in (Spring & Al-shaer, 2020)). This model can be recorded in formats to aid analysis through software (examples shown in this work) (see also Building a Model).
- Secondly, the population of this model is exercised using a representative set of examples. These attack sequences are drawn from both open-source attack reports referenced through MITRE and additionally researched open-source reports (see also Loading a Representative Data Set).
- Thirdly, several simple usage examples (implemented in Python (see (Maidens, 2023)) are presented
    - Demonstration that sequencing of attack TTPs improves the ability to distinguish between the different attack sequences recorded (often associated with APTs) (see also Using the Attack Model – LCSS Fragment Matching).
    - Demonstration of a pattern matching approach (in this case LCSS) to compare observed sequences with existing attack signatures (see also Using the Attack Model – LCSS Fragment Matching).
    - Demonstration of how data in this form can be used as input to a Hidden Markov Model (HMM) (see also Using the Attack Model – Hidden Markov Model).
    - Demonstration of a next step prediction approach (in this case Markov Model (MM) see also Using the Attack Model – Markov Model).
    - Demonstration of how relationships between ATT&CK Tactics and Kill Chain models (in this case the Unified Kill Chain (UKC)) may be studied further (see also Using the Attack Model – Unified Kill Chain).
- Fourthly, a number of Future Work proposals are made to illustrate further potential for this work (see also Future Work).

1.2 Motivation

Cyber-attacks and threats are continually adapting and developing as Cyber Space itself evolves. Despite continued development of cyber security systems the number of annual recorded cyber-attacks remains on the rise (e.g. “Cybersecurity attacks have continued to increase through the

years 2020 and 2021, not only in terms of vectors and numbers but also in terms of their impact" (ENISA, 2021)).

Openly available Cyber Threat Intelligence (OSINT or CTI) provides a valuable contribution to a Cyber Actor's ability to achieve cyber situational awareness ("cyber situational awareness is the part of situational awareness which concerns the 'cyber' environment" (Franke & Brynielsson, 2014)). This is true for individuals as well as small and large organisations.

A particular area of investigation involves the study of Advanced Persistent Threats (APT) ("APTs are cyber-attacks executed by sophisticated and well-resourced adversaries targeting specific information in high-profile companies and governments, usually in a long term campaign involving different steps" (P. Chen et al., 2014)). These complex attacks are particularly difficult to defend against.

Some of the key features of these types of attacks include the sophistication of the 'trade craft', the choice of targets and related objectives and the often extended timeframes (c.f. (Ussath et al., 2016))

Many cyber security systems provide sophisticated methods for identifying Indicators of Compromise and anomalous network traffic etc but still (understandably) struggle to defend against sophisticated sequencing of APT attacks.

MITRE (formed out of Massachusetts Institute of Technology, "not an acronym but the name of our company" (MITRE, 2019b)) has missions that include the wish 'to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity' (MITRE, 2019c).

MITRE ATT&CK (Adversarial Tactics, Techniques & Common Knowledge) is a dataset providing a globally accessible knowledge base for standardising a Malicious Agent's behaviour (TTPs) observed during cyber-attacks. These standardised descriptions are intended to encourage analysts to create greater standardisation in reporting cyber-attacks, that may be subsequently made visible through openly available (where appropriate) reports. This knowledge base is becoming increasingly popular and is indeed helping to standardise description of the Tactics and Techniques used in publicly available reporting of APT attacks.

The knowledge base also includes descriptions for several (currently over 100) significant APTs and the TTPs that they use. However, these descriptions are limited in their application as they simply list the TTPs used by the APT and do not provide more detailed intelligence on specific attacks and the sequencing of the TTPs used (e.g. p8 (Spring & Al-shaer, 2020)). There is a

potential to provide additional intelligence that can be used by defenders to detect suspicious events within their environments through the observation of suspicious sequences of TTPs that may be related to previously observed APT activity.

As stated in (Ahmed et al., 2021) much research into APT activities is constrained by a lack of data. This is also discussed in (Lemay et al., 2018) and (Alshamrani et al., 2019) where the reports available to study details of APT attacks are limited to a small number of well-known reports. It is also noted that much of this information is made available through industry as opposed to academia.

Based on the observations above, this work seeks to demonstrate an approach to improving the contribution that MITRE ATT&CK can make to an organisation's cyber situational awareness through the provision of TTP sequences ('signatures') representing previously reported cyber-attacks (that may be used by software to analyse sequences of TTPs observed in a system). It is hoped that a subsequent implementation of this proposal (as an openly available dataset) would provide a useful contribution in this area and indeed provide a platform to help coordinate subsequent collaborative research activities.

1.3 Proposal Outline

This research proposal investigates one possibility for increasing the contribution that the ATT&CK dataset can provide to a cyber analyst's situational awareness.

It will codify a representative sample of cyber-attacks from the reports included within the current knowledge base. Although some additional reports external to ATT&CK have been referenced while studying the attacks, this work has currently limited to attacks referenced within the knowledge base to ensure that the data is consistent. This codification is then presented as an accompanying knowledge base of codified sequences of techniques (and tactics). These sequences then being used to match detected/observed attack fragments to potential known sequences. This knowledge is then used to provide additional intelligence to the analysts to help prioritise subsequent courses of action.

To focus the scope of this work, the investigation is built on the assumption that a system is (or will be in the future) available that can provide candidate attack event sequences ('attack fragments') through correlated event observations within the underlying configuration.

The figure below provides a brief overview and context for the proposal investigated in this document. At this point Additional Open CTI is limited to CVE references, however future work

proposals include extending the attack model further to accommodate additional sources (see also Future Work).

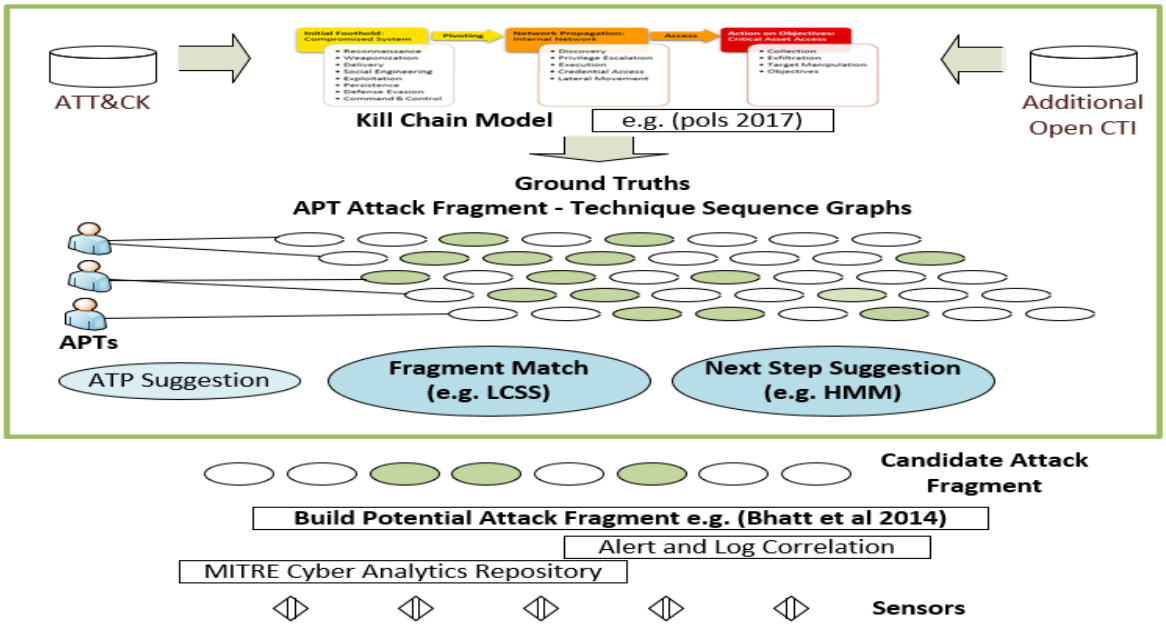


Figure 2 - Proposal Outline

One specific point of note that it does not intend to provide an approach to attribute the attacks to specific groups. This is a different question requiring additional intelligence. Matching observed attack sequences may indeed provide 'interesting' intelligence that may encourage further investigation for specific APT related behaviours and artefacts.

Given the granularity of this middle tier knowledge base, we may well find that groups have similar or even equal attack patterns. In this case it will be sufficient to match the correct pattern and use this as the basis of subsequent decision making.

1.4 Contribution

The key contributions in this work:

- A literature review laying out current state of the art in kill chain detection.
    - This is used to justify the assumption that a future system will be capable of detecting 'attack fragments' that may be used to detect fragments of sequences of tactics and techniques used in an attack.
- A proposed model to augment the existing ATT&CK dataset with attack sequence information.
    - This includes a meta data model to support attack classification
    - This also includes an approach to recording known attack sequences.

- A demonstration on how this model can aid 'classification' of an 'attack fragment'.
- A demonstration on how this model can aid analysis of likely next steps and therefore prioritisation of an appropriate course of action.
- A downloader for the ATT&CK dataset developed in python (see (Maidens, 2023))
    - This converts the data into a set of relational tables. These are then made available to other python developments through an object with helper methods.
    - Additional similar developments have been previously implemented for CAPEC, CWE and CVEs during this work but have not been used here.

1.5 Research Questions

The research questions below were derived from a much broader set of original questions investigating opportunities for improving machine readability of key MTRE and NIST open cyber threat information sources. These included considerations such as presenting these data sets as part of a Semantic Web Stack, linking changes in APT behaviours and CVE publication dates (there has latterly been some related investigation into this e.g. (Kuppa et al., 2021) (Hemberg et al., 2020) (MITRE, 2022e)) and linking MITRE ATT&CK, CAPEC, CWE and CVE datasets. This final set of questions resulted from research during these studies. This research identified benefit (as described above) from provision of tactics and technique sequences used in the cyber-attacks described in the attack reports referenced by the ATT&CK knowledge base alongside the current content (of this knowledge base).

	Main Research Questions
R1	Can the ATT&CK APT descriptions be used to support the detection of multi-step cyber-attacks and potentially anticipate next steps?
R1a	Can we record known APT attacks as sequences of ATT&CK Tactics and Techniques?
R1b	Will the sequences in R1a provide us with additional intelligence over and above the unordered lists of APT Techniques currently provided within the ATT&CK knowledgebase?
R1c	Can we provide a classification system for the sequences in R1a that will also support some further analysis of recorded attacks?
	Supporting Research Questions (Literature Reviews)
R1_SuppA	Can we create multi-step ATT&CK technique chains from sensor networks?
R1_SuppB	Can we extract interleaved attack chains from ATT&CK techniques detected from sensors?

Table 1 - Research Questions

1.6 Report Structure

The document below is structured as follows:

- Chapter 2 (see Background) provides a broad background of related areas of study.
    - This section also notes that there remains a lack of precise clarity of terms across literature and seeks to establish terms for this document.
    - This work brings together several related areas.
- Chapter 3 (see Related Work) provides a review of related subject matter to establish a view on relevant state of the art in these areas
    - It seeks to justify the assumption that we can (theoretically) sample fragments of an attack (kill chain) in a given IT environment.
    - This also identifies the gap in research relevant to this work
- Chapter 4 (see Research Questions) defines the proposal and research questions
    - It also provides a summary of general approach
- Chapter 5 (see Characterising the Base Data) provides a more detailed view of the ATT&CK dataset being studied
- Chapter 6 (see Building a Model) provides a discussion on the creation of the model proposed to record the attack sequences
- Chapter 7 (see Results) provides results obtained from example applications
- Chapter 8 (see Conclusion) provides a summary of conclusions and future work proposals
    - Here the future work proposals are important as they are part of the conclusions in terms of the limitations encountered at this point

Chapter 2 Background

2.1 Introduction

In this section I will describe the background areas of knowledge and research relevant to the broad purpose of this thesis.

It is provided to give:

- Clarity on terms used (along with relevant literature) (see Defining Basic Terms)
- An overview of available Cyber Threat Intelligence (see Cyber Threat Intelligence)
- An overview of Cyber Situational Awareness (see Cyber Situational Awareness)
- A brief summary of the rational for investigations within this thesis (see Conclusion and following Chapter Related Work).

2.2 Cyber Security as a Science

As a 'science' of Cyber Security (potentially) develops, the terms used within this discipline still require agreed and clear global definitions.

This issue is reviewed and discussed in (Ramirez & Choucri, 2016) that highlights the need for common terminology to underpin interdisciplinary collaboration and further development in this area. The paper references the 'recently dubbed' term 'cybermatics' to bring together all perspectives of a wider. The 2022 IEEE Cybermatics Congress defines this as 'Cybermatics is to build systematic knowledge about new phenomena, behaviors, properties and practices in the cyberspace, cyberization and cyber-enabled hyper worlds' (*2022 IEEE Cybermatics Congress*, 2022). Similarly in (Althonayan & Andronache, 2018) the authors highlight the need for unified terminology. Again in (Suryotrisongko & Musashi, 2019) an attempt at presenting a taxonomy of cyber security research topics is presented.

This desire to explore this situation has been explored for a number of years, for example (Introduction (Ramirez & Choucri, 2016)) states "Cyber security is a nascent and exploding field with a growing body of research" …. "[s]tandardisation is commonplace in scientific disciplines, beginning with either systematic nomenclature or otherwise standardized vocabulary"). We also have "Part of [the] problem stems from the lack of standardized methods for talking about cybersecurity concepts. This problem was called out in a report commissioned by the Department of Defense in 2010" (Applebaum et al., 2018). And again in (Lallie et al., 2020) (Section 2) the

author specifically notes “The cyber security domain suffers from two specific problems relevant to the present paper: inconsistency in the ontological terms, vocabulary, and definitions used to describe the domain and ….” .

Examples of work on refining definition of specific terms also include:

- An attempt to make the terms Tactic, Techniques and Procedures more precise to aid both human and machine understanding. (Maymí et al., 2018)
- A developing Cyber Security Body Of Knowledge (CyBOK) (Bristol, 2023).

And in a related direction the “Need for Collaborative Intelligence in Cybersecurity “ is discussed in (Martin, 2021). Here the focus is on a consistency in concepts allowing a more ‘human like’ fuzzy decision-making approach to be achieved.

Given the continued need for standardisation, I have initially tried to define some specific terms used within the scope of this document to at least aid clarity here. I have tried to provide references to relevant background research moving towards current thinking. Often these works themselves reiterate the lack of precise clarity in many of these areas.

2.3 Defining Basic Terms

In this section, I have tried to give definitions to be used in this document. Later, these will form input to concepts being modelled (see also (NIST, 2019a) (NIST, 2013)).

2.3.1 Cyberspace

“**cyberspace** consists of the globally connected networks of hardware, software and data […] that humans can interface […] and in doing so become part of it”

We (arguably) now live in a world where the traditional ‘spaces’, of Physical, Social and Thinking (see Abstract (Ning et al., 2017)), that we have occupied, have been extended and complemented by a ‘Cyberspace’ developing from the information and computer technology developments of recent years.

Arguably credited to the author William Gibson (Neuromancer,1984) the use of the word ‘Cyberspace’ is relatively recent (a readable summary sketch is given in (Wikipedia, 2019) relating the use of this term to the development of the internet).

There is continued refinement in the understanding of the meaning and context of this word. For instance in Chapter 1 (Garvey, 2021) examining the relationship between Cyberspace and the Supply Chain, Garvey recognises multiple de facto meanings of this 'new word' and appeals for the need of a 'thorough philopsophical examination' of what the word's meaning. This probably reflects society's struggle to come to terms with the nature, benefits and risks associated with this rapidly evolving concept.

The definitions here are intended to provide foundations for modelling concepts later, so for this document I will use a very simplified view "cyberspace consists of the globally connected networks of hardware, software and data […] that humans can interface […] and in doing so become part of it" (Ottis & Lorents, 1984). Even this simplified view implies 'a complex socio-technical system (of systems)' (Executive Summary (Lange et al., 2017). A hint of the rich technical and societal implications of a developing Cyberspace is given in (Ning et al., 2018) which proposes the concept of General Cyberspace (GC) integrating the generally understood view of cyberspace as a digitisation of the traditional physical, social and thinking spaces (PST) with a broader concept of a cyber enabled spaces., but developments such as this cannot be discussed in detail here. We may see a simplified view of Cyber Space as operating over three separate but related layers (Pols, 2017). An inner technical layer, supporting and enabling a socio-technical layer all of which is controlled within a governance layer. Here I am only focussing primarily on the technical layer, as illustrated below.

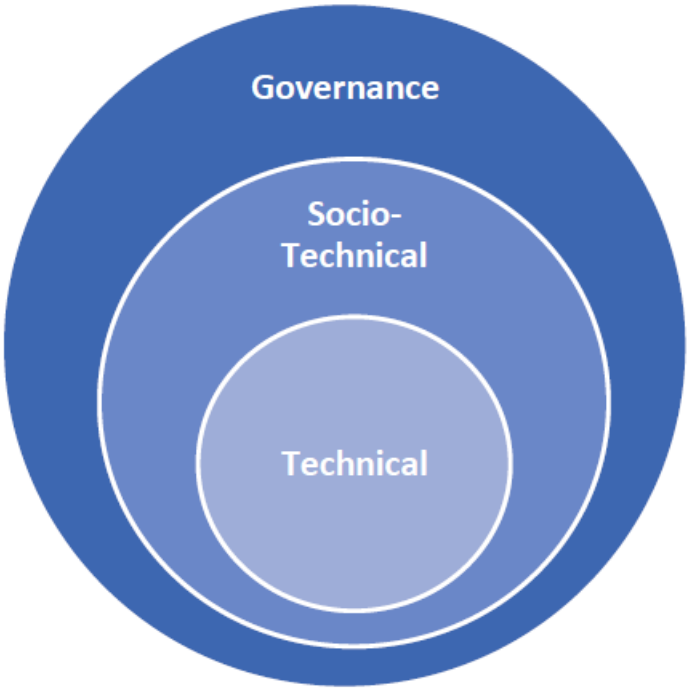


Figure 3 - Layered Cyberspace Model

(Pols, 2017)

2.3.2 Cyber Actors

Within Cyberspace there are several basic actor types that are relevant to this work.

Cyber Actor
An individual, organisation or process interacting with cyberspace

Malicious Cyber Actor (or **Threat Actor**)
An individual, organisation or process interacting with cyberspace with malicious intentions

It is common to categorise these Malicious Cyber Actors into major groupings (for instance the a quite detailed view in the Intel Threat Agent Library described in (Casey, 2018)).

Another high level 'typology' is often presented along the lines of (Andress & Winterfield, 2011)

- Advanced Persistent Threat (APT) - A threat actor (often very sophisticated and possibly state sponsored) who gains access to a Cyber Actors set of assets and can remain undetected for a significant period of time (Alshamrani et al., 2019).
- Organised Crime – Also known as Cybercriminal group
- Insider Threat – Disgruntled or otherwise motivated employees
- Hacktivist – Motivated by political views
- Script Kiddies/Noobs – Less skilled attackers using tools that can be found on the internet

A similar but slightly extended threat actor typology is again offered in the 2016 Cyber Security Assessment Netherlands (CSAN) (National Cyber Security Centre, 2016). Another typology is also offered in (Meyers et al., 2009) but for this work the above categorisation is sufficient.

Recognising some of the limitations in the structuring of this particular typology and in particular the inherent "lack of consistent dimensions for distinguishing actors' a proposal for a new typology is developed in (De Bruijne et al., 2017). This work also a recognises that any such typology needs to adjust to the dynamics of the underpinning intelligence (2.3.2 p12 (De Bruijne et al., 2017)). The resultant proposal is based around five key dimensions:

- Target
- Expertise

- Resources
- Organization
- Motivation.

This typology is used only to classify incidents and associated actor type. It does not seek to address specific attribution. Example proposals related to this specific area of investigation are provided by the Q-model presented in (Rid & Buchanan, 2015) and an application of an argumentation based reasoner to this Q-model in (Karafili et al., 2018)

In (Mavroeidis et al., 2021) an approach to automatically inferring actor types is developed. This model hopes to support an element of polymorphism within its structure to better reflect some of the complexities around the behaviours and motivations of a malicious cyber actor (e.g., a cybercriminal being hired by a nation state).

Annual cyber threat reports from cyber security companies (e.g. (Accenture, 2021) or (National Cyber Security Centre, 2021)) and indeed many other generally available attack report feeds provide an overview of approaches used by Malicious Cyber Actors.

They key interest in this works focusses mainly on the Advanced Persistent Threat (APT). A consideration of the problems, threats posed by APTs and the life cycle (e.g. “reconnaissance, initial compromise, establishing foothold and infiltration”) of these types of attacks is explored in (Rot & Olszewski, 2017). Some investigation and analysis of differences between APT / traditional attacks and methodologies is provided through critical analysis in (Siddiqi & Ghani, 2016)

Advanced Persistent Threat (APT)

A threat actor (often very sophisticated and possibly state sponsored) who gains access to a Cyber Actors set of assets and can remain undetected for a significant period of time (Alshamrani et al., 2019).

A particularly notable analysis is given in (Mandiant, 2013). This provides an analysis of APT1 (potentially a threat sponsored by the Chinese state) and was an early report describing an APT and demonstrates many of the qualities we associate with the attacks associated with this sort of an Actor.

- Technical sophistication
- Significant access to resources
- Types of targets (organisations and objectives)
- Long timeframes and campaigns
- A desire to maintain persistence over a long period of time

- Willingness to take time over each step within an individual attack.

Additional surveys and discussions of APT literature are given in (Lemay et al., 2018) and (Alshamrani et al., 2019). A fuller (or at least longer) description is provided in (NIST, 2023) and (H. Kim et al., 2019) (section 2.2) but the APT definition provided above seems to capture the key points, except perhaps the observation that the APT may adapt to a defenders capability over a period of time. A recent survey of concepts and difficulties associated with detecting and understanding APT attacks is provided in (Khaleefa & Abdulah, 2022). This includes an outline of a typical APT lifecycle initially proposed in (Quintero-Bonilla & del Rey, 2020). It also describes key characteristics (derived from previous publications) of the APT.

An attempt at classifying various nations cyber-criminal activity according to various typologies is provided in (Kigerl, 2016) . The conclusions reached here are based on very specific threat types so final decisions on cluster types may be debatable. A nation index of cyber 'criminality' is reviewed in (Lusthaus et al., 2020).

However, these evolving works and continued research suggest that there remains potential for further investigation into evolution of an APT typology/classification model.

2.3.3 Cyber Actors and Cyber Assets in Cyber Space

As noted above "**cyberspace** consists of the globally connected networks of hardware, software and data […] that humans can interface […] and in doing so become part of it". That is a Cyber Actor interacts (interaces) with Cyber Space through Cyber Assets.

Cyber Asset

Hardware, software, or data (that may itself be networked) and connected to Cyber Space.

The ability for a Cyber Actor to clearly identify the assets they are using during this interaction is potentially an extremely complex process. For complex organisations the relevant assets may be difficult to specifically identify and itemise (e.g., cloud-based implementations). Even where assets can be clearly identified there is the issue of constant change of assets and their interdependencies. Some example approaches to support a Cyber Actor's identification of their Cyber Assets include in the "Security Guideline for the Electricity Sector : Identifying Critical Cyber Assets" (NERC. North American Electric Reliability Corporation, 2010) , the NIST Asset Reporting Format (ARF) (Halbardier et al., 2011) and associated (Wunder et al., 2011).

Clearly a Cyber Actor interacting within Cyber Space creates the potential for interaction with other Cyber Actors (also within the globally connected Cyberspace). These interactions may be welcomed and beneficial or may be unwelcomed and have malicious intention or unintended results.

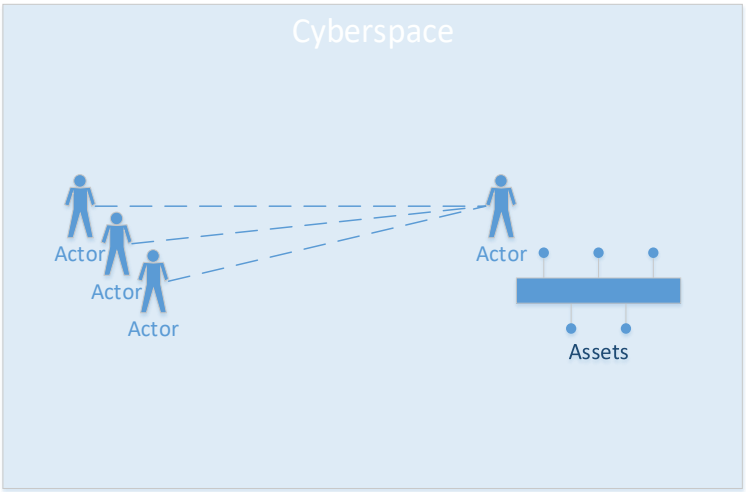


Figure 4 - An Actor in Cyberspace

2.3.4 Malicious Actors - Tactics, Techniques, Tools & Procedures

The activities of malicious threat actors can be described in terms of Tactics, Techniques and Procedures (TTPs). These terms have come from the military, but these are not as well defined in cybersecurity. An attempt to move towards a more precise definition and formal semantic modelling that may ultimately better interact with machine learning approaches is provided in (Maymí et al., 2018). The modelling is exercised using intelligence from a specific APT28.

Some key modelling elements from this paper are summarised below:

- A **task** is as a clearly defined action or activity specifically assigned to an individual or organization.
- An **objective**, which is a clearly defined, decisive, and attainable goal toward which an operation is directed.
- A **procedure** is a fixed, ordered, complete sequence of primitive actions.
- If something looks like a procedure, but it is possible to omit or reorder steps, then it is actually a **technique.**
- A **tactic** consists of techniques (together with tools)

An earlier and more generalised approach was provided in Ryan Stillion's seven layer model (stillion, 2014). This provides some intuitive hierarchy of context to the approaches and

motivations of attackers and how these may be evidenced. This is shown below and moves downward in terms of specificity/generality providing a visual and instinctive summary of the concepts.

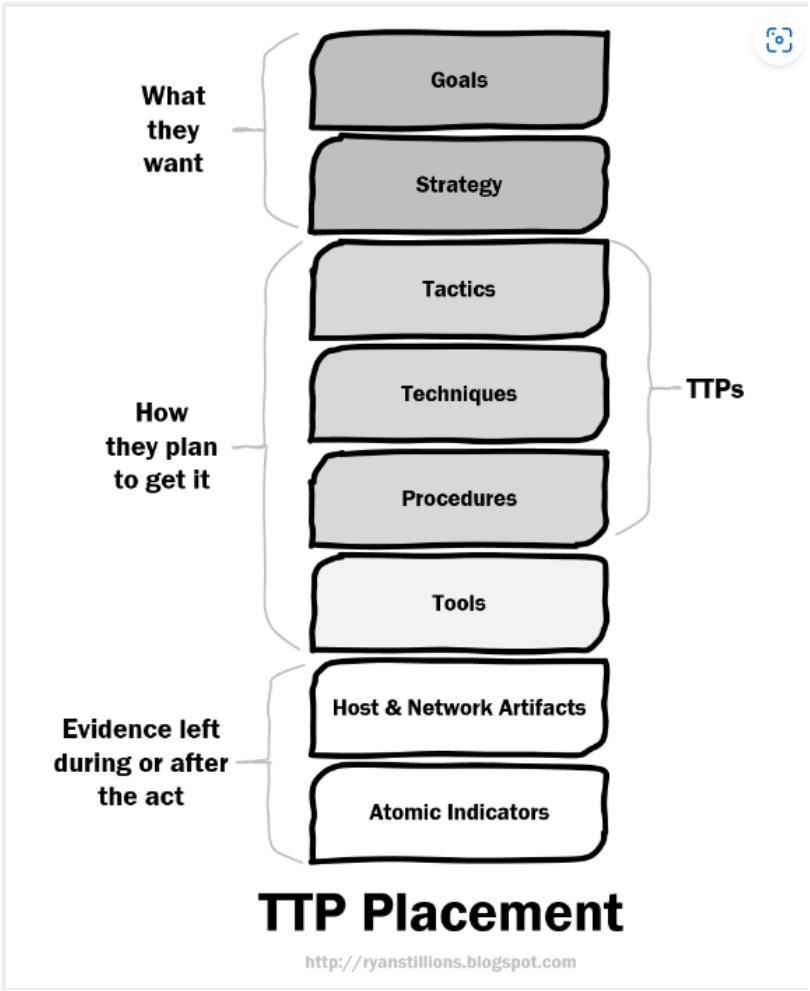


Figure 5 - Stillion's TTPs (stillion, 2014)

The modelling presented here was also designed to integrate with the MITRE ATT&CK model (this model is described in more detail below). MITRE ATT&CK does not explicitly formally define these concepts as attempted in this paper however they are implicit in the model created.

Related work on Semantic Cyber Threat Modelling is presented in (Bromander et al., 2016). This builds further on both the Stillion and ATT&CK models and reiterates the need to build semantic models in order to increase the effectiveness of machine-based analysis. This also suggests the need to start to structure available cyber threat intelligence to support these ends.

2.3.5 Threats, Risks and Vulnerabilities

The benefit provided by the development of Cyberspace has also created new threats that must be recognised and managed as risks. Amongst these threats there exists the potential for a 'system' to be compromised by malicious actors.

This is not a new phenomenon, indeed from the very beginning of telecommunications there have been examples of malicious intervention. Take for example the famous example of Nevil Maskelyne's hack of an early demonstration of wireless communication by Guglielmo Marconi in 1903 (Marks, 2011).

Cyber **vulnerability**

A weakness in an IT system [Cyber Assets] that can be exploited by Malicious Actors. They can occur through flaws, features or user error, and Malicious Actors will look to exploit any of them, often combining one or more, to achieve their end goal (based on (NCSC, 2015)).

Cyber **threat**

"Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service" (Wynn et al., 2011)

The above definition is quoted from the MITRE Threat Assessment & Remediation Analysis (TARA) (Wynn et al., 2011). This approach investigates the susceptibility of a Cyber Actor's system to attack. A 'Threat Susceptibility Matrix' can be produced that considers potential known Tactics, Techniques, Tools and Procedures (TTPs) that may be used by a malicious Cyber Actor and how the risk presented by these TTPs affects the Cyber Actor's assets.

The specific focus here is on defending against an attack from an APT.

Cyber **Attack**

A set actions taken by malicious Actor(s) to successfully realise a threat though vulnerabilities.

2.3.6 Threat Modelling

2.3.6.1 Understanding Risks

If an actor wishes to understand more about the risks they face and manage them, they may choose to model the threats and work out how these relate to the understood vulnerabilities.

In general, a diligent Actor will wish to consider:

- Likely threats to which they/their Assets may be subjected.
- What vulnerabilities exist in their Assets.
- What attacks may be expected (and how likely are they).
- What impact would there be if a Threat exploited a Vulnerability.

Threat modelling

A process by which a cyber actor attempts to understand cyber threats and vulnerabilities that are relevant to them. Using this to develop approaches to mitigating the observed risks

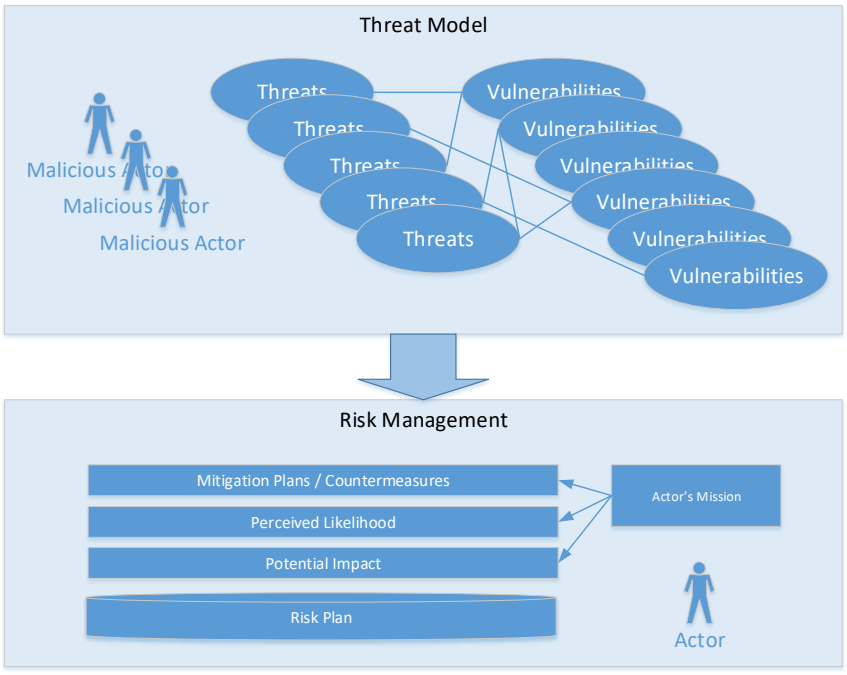


Figure 6 - An Actor's Threat Model & Mitigation Approach

Cyber threats can represent risk to a Cyber Actor.

Cyber risk

For a particular Cyber Actor, a cyber risk is the risk that a specific cyber threat represents for that actor.

Mission
Priorities of the Cyber Actor's purpose. The mission will contribute to a Cyber Actor's appetite for risk.

An Actor may choose to mitigate or tolerate risks subject to their mission (the priorities of the organisational purpose).

There is much written on the management of risks. One well known standard text giving a broad overview is provided in "Threat Modelling Designing for Security" (Shostack, 2014). But for this work, only the simplified view above is required.

Key features of a risk generally include understanding of likelihood; impact should the threat occur; overall quantification (to aid prioritisation); treatment (e.g., accept or avoid); mitigation approach and action plan should the risk become an issue. It is important to be clear the 'understanding' of risk is often subjective and represents an agreed position accepted within an organisation. Over and above documentation of a view on risks, an approach for reviewing and developing this understanding must also be agreed.

Analysis of risks can then be used to design a relevant risk management strategy.

2.3.6.2 Threat Modelling Approaches

"The use of a threat model has long been the foundation of a robust security process" is given as key principle in while considering MITRE ATT&CKs role in a threat based security approach - Section 2.3 (Strom et al., 2017). In this section I intend to provide just a brief overview of general approaches to illustrate key threat modelling approaches.

Wider views are given in (Lange et al., 2017) (Myagmar et al., 2005) and a good overview of approaches currently commonly used, is provided by Jeremy Straub in the Background section of (Straub, 2020).

A systematic overview of general threat modelling approaches and research is provided in (Xiong & Lagerström, 2019). Such approaches not only contribute to security evaluation and mitigation of a cyber actor's system configurations but also to the development of secure applications. The authors identify research relating to manual, graphical and formal approaches. Less commonly they identify research into automatic approaches. This perhaps relates to the sheer complexity of such an objective. This work also highlights the development of approaches specialised to system and attack types. One specific system area of current interest is that of Industrial Control Systems

(ICS). MITRE ATT&CK (expanded further below) differentiates between Enterprise, Mobile and ICS views of TTPs. This paper also finds threat modelling to be a wide and complex field lacking common definitions.

APT style attacks are discussed in (Tatam et al., 2021). The paper attempts to identify existing gaps in APT attack modelling and subsequently how APT attack modelling can be improved. The paper identifies four main threat modelling approaches. These are asset, system, threat, and data-based approaches. These are illustrated in the diagram below. The paper finds that it is rare for a single approach offers a definitively robust analysis and in most cases a strategy using elements across all these approaches offers the most effective path. Specifically, it recommends using 'correlated and actionable threat intelligence from multiple internal and external sources'. Additionally different approaches should be considered to communicate to different audiences.

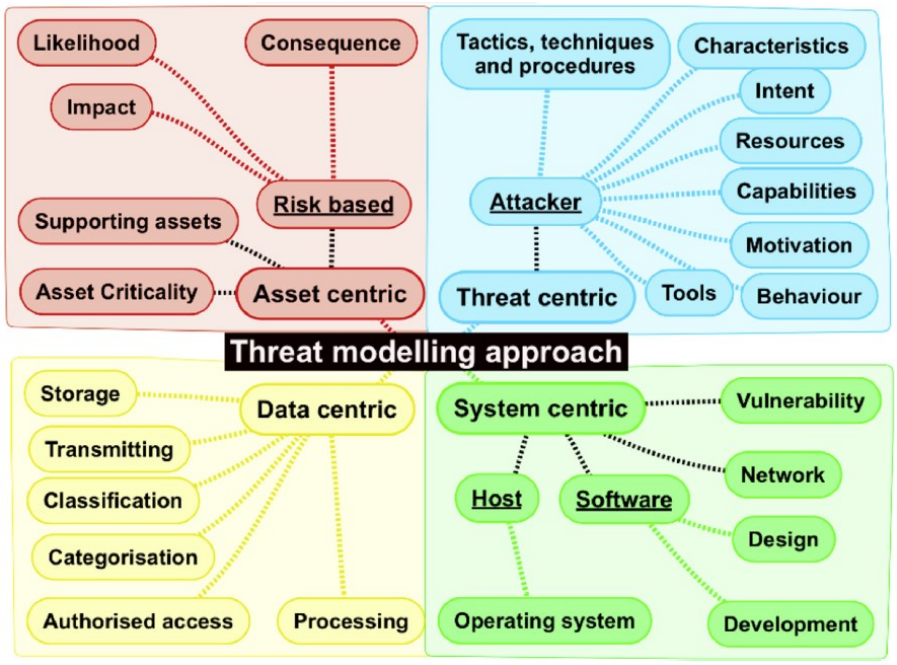


Figure 7 - Threat modelling approaches (Tatam et al., 2021)

A more recent review of visual approaches to attack modelling is provided in (Lallie et al., 2020).

An example of a consideration of appropriate approaches is given in a comprehensive survey (by the US Homeland Security Systems Engineering & Development Institute) of modelling approaches given in (Bodeau et al., 2018b). This provides a good overview and review of practices (at the point of writing). This survey was conducted to develop a suitable approach to support the work of the US government Apex program. The report found that existing frameworks required tailoring to the specific needs of the program's objectives and provides a practical example of the difficulty in developing a truly generic approach.

Domain Specific Languages (DSL) relevant to threat modelling have been investigated to potentially aid in the inevitable tailoring activities. For example, in (Yi et al., 2013) we have an enterprise threat modelling language presented. This is based on the MITRE Enterprise ATT&CK matrix. Here the objective is to measure the resilience of the system to threats. This builds on the Meta Attack Language (MAL) framework presented in (Johnson, Pontus; Lagerstrom, Robert; Ekstedt, 2014) (and further developed in (Johnson et al., 2018)).

Some examples of varied major and commonly used generalised approaches (also described in (Shevchenko et al., 2018)) are briefly summarised below:

- STRIDE and threat modelling (an architectural approach "designing for security") is discussed in (Shostack, 2014)(Microsoft, 2022)
- PASTA (Process for Attack Simulation & Threat Analysis) is presented in (Ucedavélez & Morana, 2015)
- A model of intrusion analysis (Diamond Model) is presented in (Caltagirone et al., 2013)
- Privacy threat modelling for software architectures is provided at (Linddun, 2022)
- A hybrid threat modelling approach is discussed in (Mead et al., 2018)
- A quantitative threat modelling and risk assessment approach is discussed in (Potteiger et al., 2016)
- Attack Trees are introduced in (Saini et al., 2008)

Finding Cyber Threats with ATT&CK-Based Analytics (Strom et al., 2017) presents a methodology to detect attacks post-compromise. The important point made here is that even with a well thought out defensive posture designed using the methodologies describe above, the ability of an attacker to compromise a cyber actor's assets should be assumed. An approach is presented based on the MITRE ATT&CK framework, helping defenders identify and build capabilities to sense and detect attacker's behaviours.

A commonly used approach to rationalising the output from Threat Modelling is using Attack Trees (see also above). This is based on a concept put forward by Bruce Schneier.

Attack Trees/Graphs

Attack Trees are conceptual diagrams of threats on systems and possible attacks to reach those threats." (Saini et al., 2008).

The basic concept can be illustrated visually with a very simple example (from (Schneier, 1999)). Here the target 'asset' is at the top and 'techniques' at the bottom.

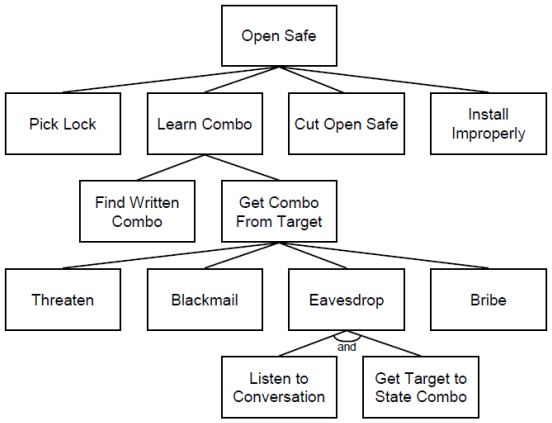


Figure 8 - Attack Tree

There have been various attempts at creating attack trees automatically using base threat intelligence. An example is given in (Aksu et al., 2018) (in this case from NIST NVD) and an overview of other generation and visualisation approaches is discussed in (Yi et al., 2013).

An extension to Attack Trees is presented in "A context-based detection framework for advanced persistent threats" (Giura & Wang, 2012). This is termed the Attack Pyramid. This places the attack goal at the top with planes allowing attack events associated within common areas to be noted. This intended to create extra flexibility in organisation and representation. A diagrammatic representation from this paper is shown below.

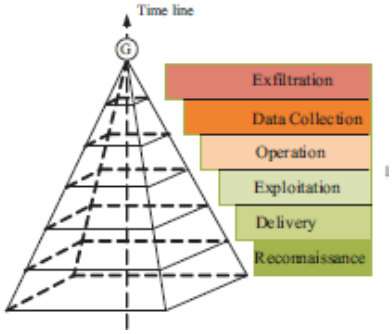


Figure 9 - The Attack Pyramid

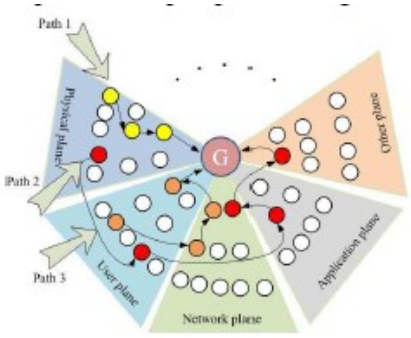


Figure 10 - The Attack Pyramid Unfolded

2.3.7 The Kill Chain

2.3.7.1 Introduction

Complex threats (for instance APT type attackers) may pass through several steps and stages before achieving their objectives.

Through analysis of the approaches used by various attackers we can understand these steps and stages (along with the appropriate indicators). This becomes an important source of analytical context that can assist an Actor in managing and identifying these complex threats. This provides an important contribution to a defender's ability to move beyond simple mitigation of known vulnerabilities to developing approaches for detection and mitigation of active successful incursions.

Behavioural modelling of attacker's behaviours based on observation and historical attacker reports has led to the creation of a number of high-level frameworks (described variously, but for example (Quintero-Bonilla & del Rey, 2020) (Ahmed et al., 2021) amongst many) to describe the key elements of these approaches. These frameworks vary in emphasis and the number of stages used to communicate the key elements of an attack.

In this section I will outline several the key relevant proposals.

2.3.7.2 Developments and Critiques

An initial framework for describing APT attacks is given in the Lockheed Martin **Cyber Kill Chain (LMCKC)** (Hutchins et al., 2011) (Lockheed Martin Corporation, 2015).

Here the author breaks intelligence indicators that may be observed during an attack into three main types.

- Atomic - Atomic indicators, e.g., IP addresses, email addresses, vulnerability identifiers.
- Computed - Computed indicators, which are derived from data extracted and associated with incidents
- Behavioural - Behavioural indicators are collections of computed and atomic indicators, perhaps best seen in terms of Tactics, Techniques and Procedures.

A seven-step model is presented to describe a generalised cyber-attack typical of an APT. This breaks down the phases of an attack into logical major steps. These phases provide a broadly sequential view of a framework that an attacker may move through while executing an attack.

- Reconnaissance

    - "Research, identification and selection of targets"
- Weaponization
    - Creating the tools, exploits and infrastructure to be used in the attack
- Delivery
    - "Transmission of the weapon to the targeted environment"
- Exploitation
    - Exploitation used to activate the weapon (e.g., a vulnerability or simply user exploitation).
- Installation
    - "Installation of [attacker assets] on the victim system allow[ing] the adversary to maintain persistence inside the environment"
- Command & Control (C2)
    - How the attacker controls the weapons and guides the objectives
- Actions On Objectives
    - Actions to achieve the objectives

The principle of the Cyber Kill Chain has been highly influential in understanding, documenting, and tackling the different stages in a Cyber Attack. A discussion on the technical methodologies, techniques and tools used in each phase is provided in (Yadav & Mallari, 2016). The kill chain model has also been used to develop taxonomies of malicious tools and software. An example is given in (Dargahi et al., 2019) where a cyber kill chain based taxonomy of crypto ransomware is developed. It should also be noted that the tools documented within the ATT&CK knowledge base are also described in terms of the ATT&CK Techniques and Tactics.

Subsequently and naturally, over time there have been a number of refinements and critiques presented (a number of these critiques are outlined in (Khan et al., 2018) (34.2)) .

A first general challenge lies around the intrusion and perimeter centric assumptions within the philosophy of the Cyber Kill Chain ((Engle, 2014)) . A view of this challenge is noted and outlined in a blog post by Koilpillai (Koilpillai, 2019). Here an example discussion is provided on how Software Defined Perimeters (SDPs) can address this challenge. A summary definition of SDPs is provided in (Cloudflare, 2022) "A software-defined perimeter (SDP) is a network boundary that is based on software, not hardware. SDPs can be part of a Zero Trust security approach". The author demonstrates (through a series of posts) how the SDP can address the seven steps of the kill chain above. However, here we will assume that these are approaches are not yet ubiquitous or perfect and that there is still value in understanding if a breach has been achieved.

A second general challenge is that the "The existing kill chain model in the IS field is problematic in that it cannot fully express the actions that occur inside an organization" (H. Kim et al., 2019) (specifically those attacks carried by an 'Insider'). This paper investigates limitations of the standard kill chain and presents a revised model that addresses internal threats within multi-media service environments. A brief article in (Greene, 2016) also considers this potential challenge band notes that once an attacker has gained a level of persistence they then effectively become an insider. This article also highlights the need for defenders to 'operate under a presumption of a breach'.

A third challenge questions how a generic attack model can be applied successfully to a variety of situations and still provide the level of detailed information/knowledge required to be useful. (Bullough et al., 2017) notes that none of the standard kill chains "address enterprise, mobile and industrial control systems (ICS) threats" and then offers up a proposal for an adaptive kill chain (Polymer) that can provide the required insight. Approaches to dealing with multi-step attacks are also required to be able to study a complete end to end sequence of steps through an organisation. In (Five, 2011) the authors explore a 'circular' form of kill chain that models the repeated steps taken by persistent attackers such as APTs as they move through the organisation.

And a fourth challenge (noted in (Choi et al., 2020) (2.2) and derived from (Strom et al., 2020) (4.1.3)) notes that the Cyber Kill Chains (and other approaches such as Microsoft's STRIDE) are high level. It does not help explain the detailed actions taken by the attacker, how these relate to one another as a sequence and the relationship between these actions and the ultimate objectives. This means that they do not directly help organise an appropriate defence and response. More detailed mid-level models such as MITRE ATT&CK assist with this.

Several example models developing the original LMCKC are explored in the following paragraphs. The focus here remains on generalised models rather than application specific examples; however some specialised examples are shown where relevant.

A high level view of a typical sequence of an APT attack is included in (P. Chen et al., 2014). Here the sequence is abstracted to six major steps

- Reconnaissance and Weaponization
- Delivery
- Initial Intrusion
- Command and Control
- Lateral Movement
- Data Exfiltration

Similarly in (Nachreiner, 2015) Nachreiner discusses some ‘tweaks’ to the LMCKC. He notes that to be of use (to a ‘Defender’), each of the phases should be matched by actionable steps that can be taken by these ‘Defenders’. Based on this he removes the Weaponization phase. He also adds Lateral and Movement and Pivoting dimensions to the C&C phase. Shown in the diagram below

This allows for a little more detail in the description of the behaviour of the attackers after the initial access and begins to describe the internal elements of a multi-step attack.

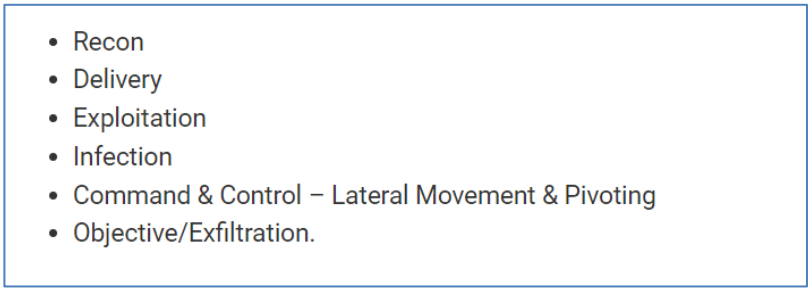


Figure 11 - Nachreiner Kill Chain

A discussion on a modified kill chain focussing on defending against a malware attack is presented in (Laliberte, 2016). As above, Laliberte argues that the Weaponization phase of the LMCKC is not required since this cannot be readily detected and adds a ‘Lateral Movement’ phase between the C&C and Actions on Objectives phases (which can be detected).

This provides some insight into the actions of the attacker once initial access has been achieved.

Figure 12 - Laliberte Kill Chain

In (Bryant & Saiedian, 2017) another attempt is presented to provide a little more detail on the behaviour of the attacker once initial access has been achieved (this time with an emphasis on

network forensics and SIEM detections).

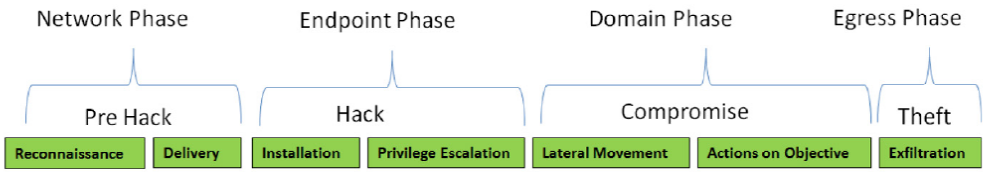


Figure 13 - Bryant Kill Chain

Together with subsequent work in (Bryant & Saiedian, 2020) this develops a framework to link SIEM detections to a kill chain model.

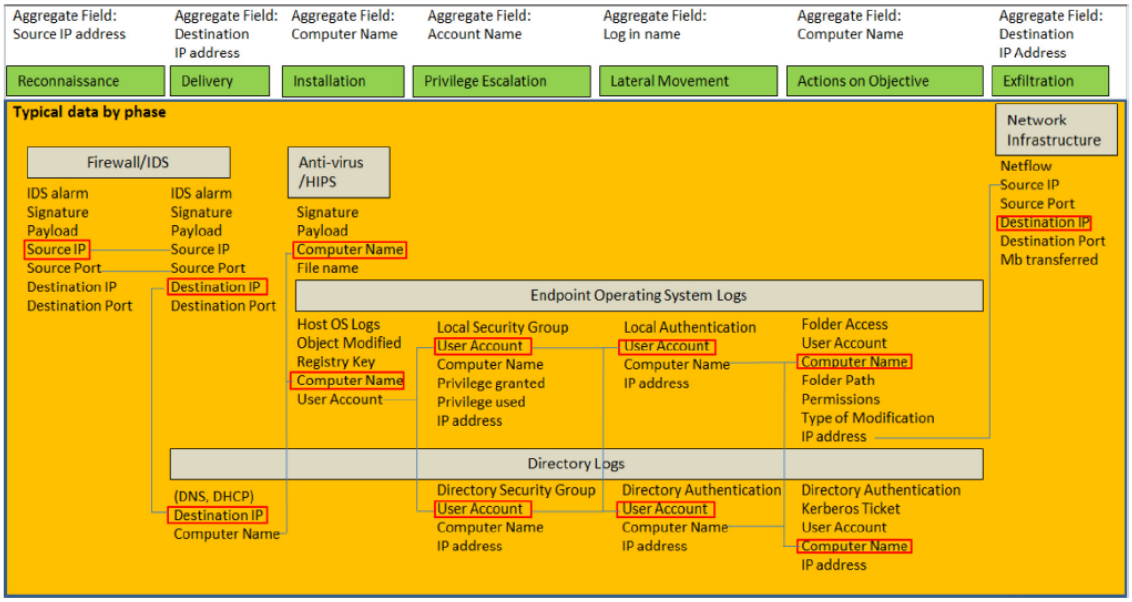


Figure 14 - Data Relationships for Correlation (Bryant)

This also provides one example of a move to create models that link sensor detections to a kill chain model to help analysts know how to organise responses and mitigations in the face of overwhelming or incomplete intelligence.

In a survey focussed on cyber scanning we have another 'anatomy' of a cyber-attack (Bou-Harb et al., 2014). This is shown below

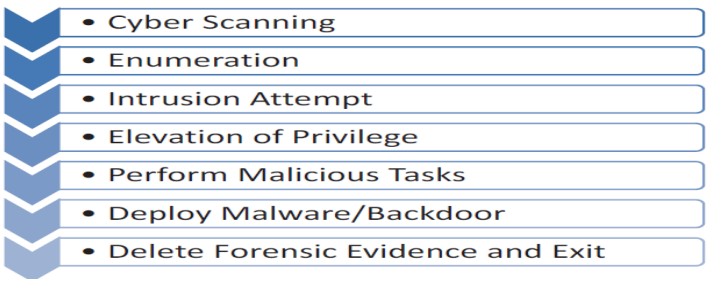


Figure 15 - Bou Harb et al Anatomy of a Cyber Attack (Cyber Scanning)

We can still broadly map this characterisation onto the models described previously, treating Cyber Scanning as a specific Reconnaissance and the following elements relating to steps of an example attack.

In (M. Li et al., 2016) the authors analyse a number of attacks included within APTNotes (Bandla & Westcott, 2019) and propose a condensed four stage model of

- Prepare
- Access
- Resident
- Harvest.

The Prepare stage, relates to the cyber kill chain reconnaissance and weaponize stages. The access stage relates to the deliver and exploitation stages. The resident stage relates to the installation and command and control stages but also explores and encompasses lateral movement, while the harvest stage relates to command and control and actions on objectives stages (although not explicitly stated, this includes the motivation of damage).

In (Ghafir et al., 2018) (following (Ghafir & Prenosil, 2016)) the author proposes an approach to automated APT detection. A set of detection modules a developed within a machine learning event correlation framework. These are linked to a generalised view of APT attack steps. These are illustrated below. Four of these steps are deemed detectable.

The MLAPT detection modules for each APT step.

APT step	Detection modules
Step 1 Intelligence gathering	This initial step includes a passive process which cannot be detected through network traffic monitoring, so there are no detection modules.
Step 2 Point of entry	Disguised exe file detection Malicious file hash detection Malicious domain name detection
Step 3 C&C communication	Malicious IP address detection Malicious SSL certificate detection Domain flux detection
Step 4 Lateral movement	This is internal traffic within the target's network. MLAPT monitors the inbound and outbound traffic, so there are no detection modules.
Step 5 Asset/Data discovery	Scanning detection
Step 6 Data exfiltration	Tor connection detection

Figure 16 - Ghafir - Generalised APT Steps

The FireEye (was Mandiant) kill chain, initially described in a report on APT1 (Mandiant, 2013) , tries to describe the movement of an attacker once an initial foothold is achieved. This can be

seen as adding a little more detail to the Laliberte suggestion above. Here a cycle of repeated activities a performed to describe how an attacker may move on from an initial entry point moving laterally through the target organisation until the mission is completed (or halted).

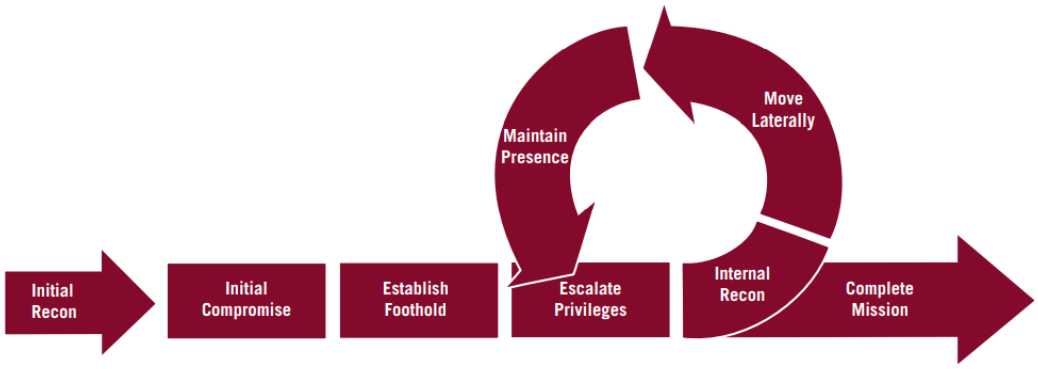


Figure 17 - Mandiant Kill Chain

An analysis of stages of an APT attacks is described in a survey on APTs provided in (Alshamrani et al., 2019) . The survey includes a diagrammatical representation of these stages in the form of a high-level attack tree and is shown below. The rectangles represent the stages.

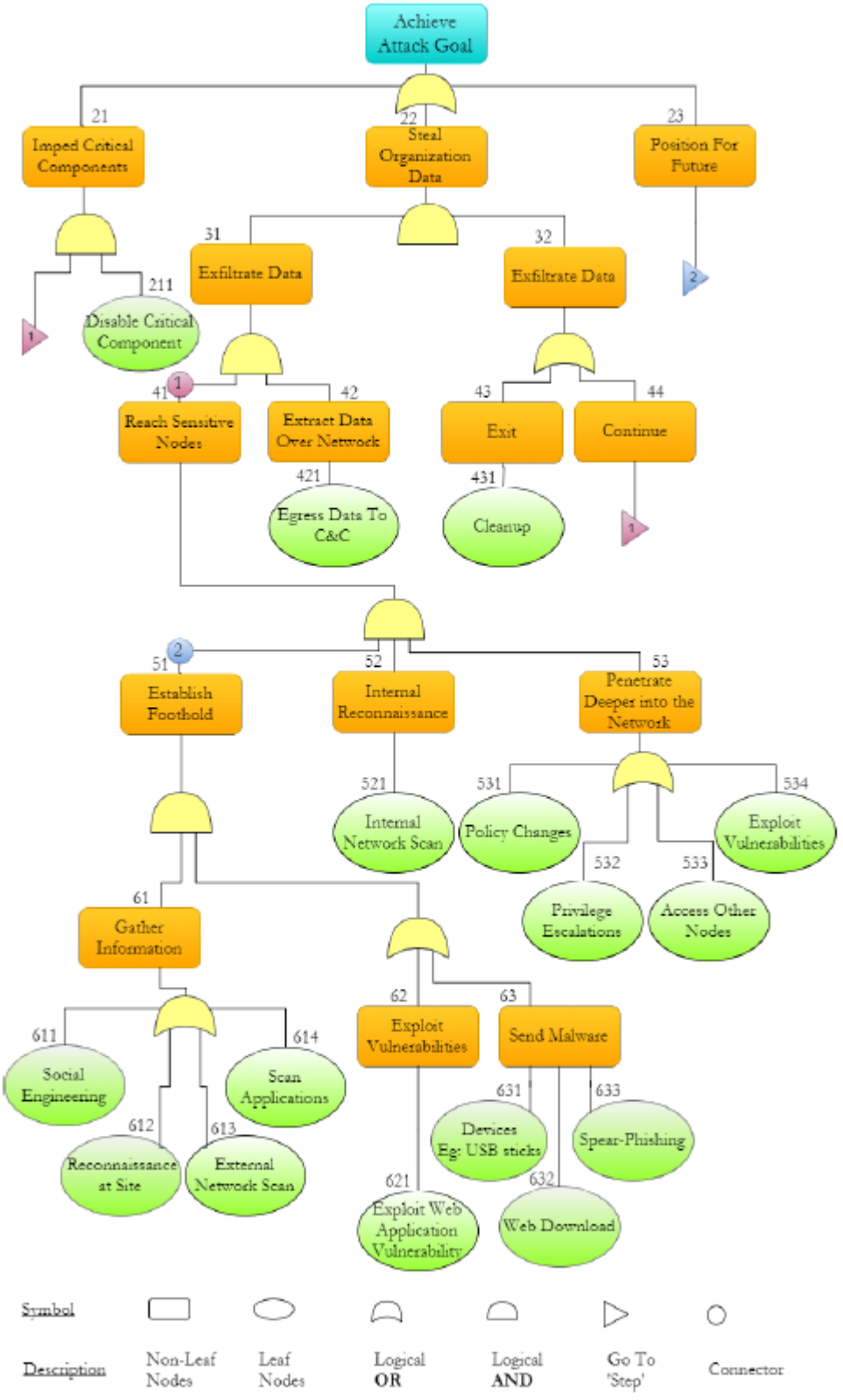


Figure 18 - APT Stage Attack Tree (Alshamrani et al., 2019)

This is further summarised as follows (subsequently the stages are related to relevant vectors of attack):

1) Reconnaissance – As above.

2) Establish Foothold - This stage represents their successful entry into their target's computer and/or computer network.

3) Lateral Movement – Moving deeper into the organisation's systems.

4) Exfiltration – When targeted.

5) Cover Up – Effectively maintaining presence by evading detection.

In (Varonis, 2018) an eight phase development of the LMCKC by Varonis is presented.

Figure 19 - Varonis Kill Chain

Again, this aims to develop a little more detail around the behaviour of the attacker once they have achieved initial access. It specifically adds elements around obfuscation/anti- forensics (which can be seen as expanding on the LMCKC Installation/Persistence phase) and lateral movement phases similar to other examples above. The circularity of the model reminds us that the attackers may repeat cycles as they move through the organisation.

In (Sexton et al., 2015) 'Attack Chain' detection is considered. This considers attack detection through combination and statistical analysis of events detected across the various steps of an attack (that may otherwise be considered independent). This is based around a five-step model that is pictured below

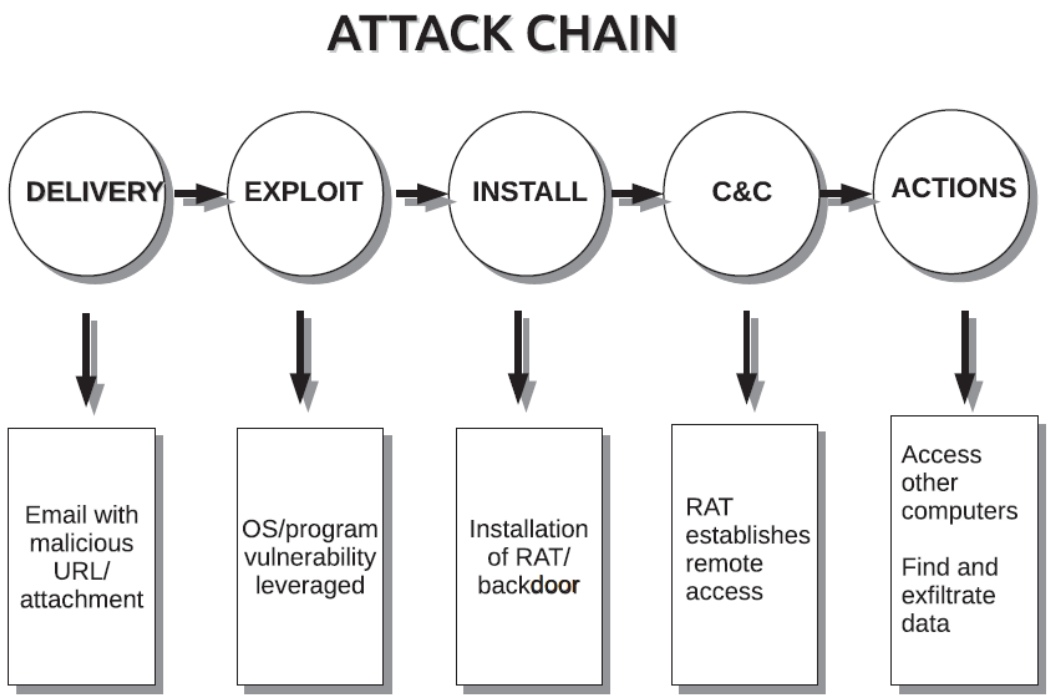


Figure 20 - 5 Stage APT Attack Model (Sexton et al., 2015)

The Reconnaissance and Weaponization phases are missing presumably, as argued variously, because these cannot easily be detected. Multi-step and lateral movement elements have been reduced to consideration under the Actions phase. But this model is used to present the approach to detecting events through multiple sensors and data sources.

Recognising that the Cyber Kill Chain is actually 'a circular and non-linear process', an extended view of the Cyber Kill Chain is summarised in (Panda Security, 2017) integrating external and internal phases of multi-step attacks using multiple kill chains. This is illustrated in the illustration below (see also Malone Kill Chain (Malone, 2016))

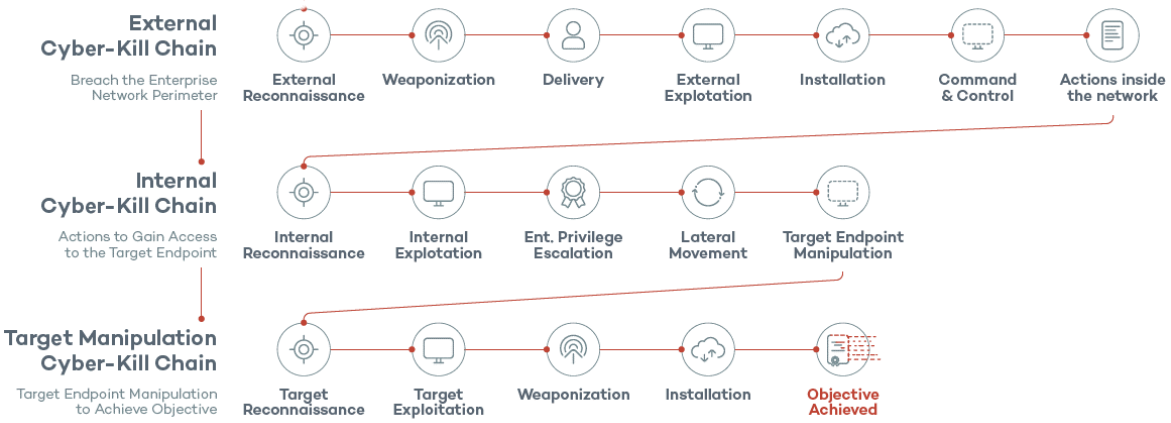


Figure 21 - Panda Kill Chains (Panda Security, 2017)

The Polymer Adaptive Kill Chain (Neto et al., 2021) aims to provide a kill chain that can be used across multiple heterogeneous environments. It consists of 18 'phases' derived through consolidation of the various ATT&CK matrices (Enterprise, ICS and Mobile). This is illustrated below.

Init	Pos	Name	Precedence	Reference	Init	Pos	Name	Precedence	Reference
RE	1	Reconnaissance	-	Enterprise	CC	10	Command and Control	1-3, 10, 11	All
RD	2	Resource Development	-	Enterprise	LM	11	Lateral Movement	3, 8-11	All
IA	3	Initial Access	1, 2, 8-11	All	RS	12	Remote Service Effects	1, 2, 3	Mobile
EX	4	Execution	3, 10, 11	All	NE	13	Network Effects	1-6	Mobile
PE	5	Persistence	3, 10, 11	All	CO	14	Collection	3, 8-11	All
PR	6	Privilege Escalation	3, 10, 11	-ICS	EF	15	Exfiltration	8, 9, 12-18	-ICS
DE	7	Defense Evasion	3, 10, 11	All	IR	16	Inhibit Response Function	3-6, 10, 11	ICS
CA	8	Credential Access	1-3, 10-18	-ICS	IP	17	Impair Process Control	3-6, 10, 11	ICS
DI	9	Discovery	3, 10, 11	All	IM	18	Impact	3-6, 10, 11	All

Figure 22 - Adaptive Kill Chain

The precedence column here is used to indicate that at least one of the phases noted must have occurred before the relevant phase can occur.

As an example of are refinement of a Kill Chain to tailor for a specific environment, (Maynard et al., 2020) presents a sequential analysis of attacks on Industrial Control Systems. It is noted that a refinement of the LMKC was required for these systems. This was developed by SANS and named the ICS-KC and is detailed in (Assante & Lee, 2015). It is outlined as follows

(a) Stage 1: Cyber intrusion preparation and execution	
1. Planning	Reconnaissance
2. Preparation	Weaponization
	Targeting
3. Cyber intrusion	Delivery
	Exploit
	Install/Modify
4. Management and enablement	C2
5. Sustainment, entrenchment, development, and execution	Action
(b) Stage 2: ICS attack development and execution	
6. Attack development and tuning	Develop
7. Validation	Test
8. ICS attack	Deliver
	Install/Modify
	Execute ICS attack

Figure 23 - ICS-KC (refinement of LMCKC for ICS)

A general refinement of particular note is presented in (Pols, 2017) where an attempt at presenting a unification of the various kill chains is presented. The design of this kill chain is derived through analysis of several APT attack reports. From these reports standard attack paths are derived and from these high-level tactics ('tactical repertoire' across the APTs examined) are recorded. This is then used to propose and justify an end to end 'unified' kill chain. The eventual chain presented is as follows:

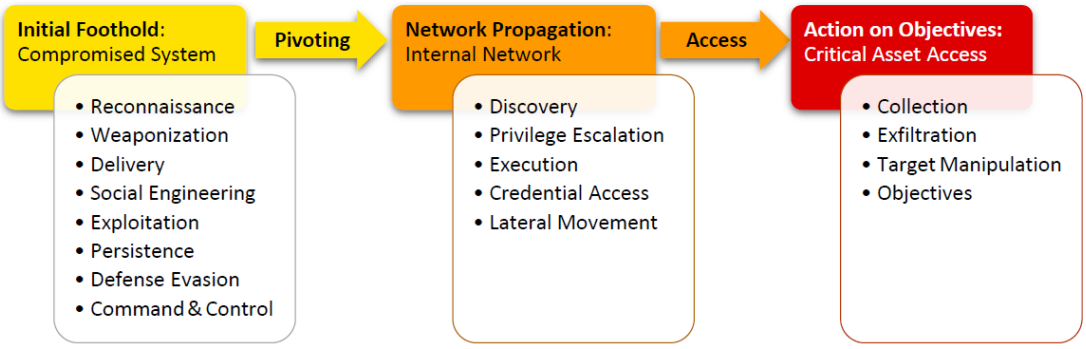


Figure 24 - Unified Kill Chain (Pols, 2017)

Here there are three major steps Initial Foothold -> Network Propagation -> Action on Objectives with supporting tactics. Here Pols makes a very specifically differentiates between Lateral Movement and Pivoting (often used synonymously elsewhere). Here Pivoting describes a position used by an attacker that can then be used to coordinate subsequent movement (e.g. Lateral Movement). For instance, after Lateral Movement an attacker may continue to use Command & Control capability established in the Initial Foothold.

In consideration of a 'Cognitive and Concurrent Cyber Kill Chain Model' (Khan et al., 2018) the potentially non-linear nature of these attacks is investigated. This is also presented alongside (Ju et al., 2020) MCKC: a modified cyber kill chain model for cognitive analysis within Enterprise multimedia network.

It is argued that the Cyber Kill Chain phases can be combined into following four categories:

- R – External/internal reconnaissance for exploitation
- D – Delivery of attacker payloads
- P – Establish persistence in the target

- CnC – Command and control across the attack (communications in and out)

This model being used repeatedly over time to model the activities of the attacker. This is illustrated alongside the Laliberte kill chain below.

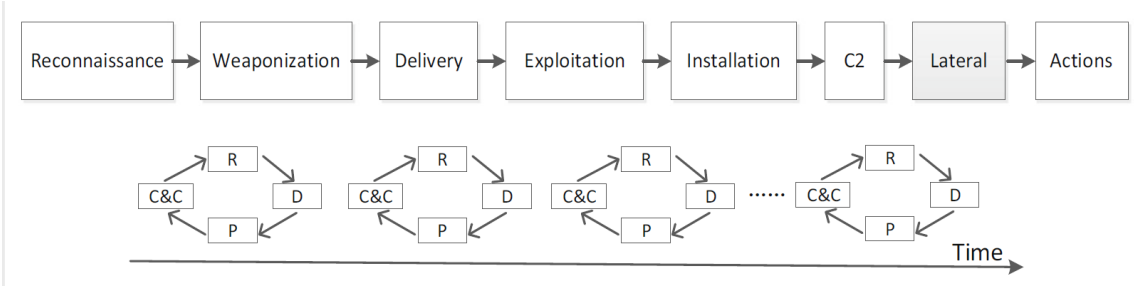


Figure 25 - Khan Kill Chain alongside Laliberte (Ju et al., 2020)

In (Hoffmann, 2019) the author argues that there has been little investigation into stochastic models of cyber kill chains. The paper assumes that the next kill chain step taken by an attacker only depends on the current step (the Markov Property "the memoryless property of a stochastic process"). It also assumes that the time spent in each state independent (and distributed exponentially) and that the probabilities of transitioning between these states are unchanging over time. In this way a stochastic model of a Cyber Kill Chain is modelled as a homogeneous Continuous-Time Markov Chain (CTMC).

This leads to a state transition diagram (for a simple LMCKC) as shown below

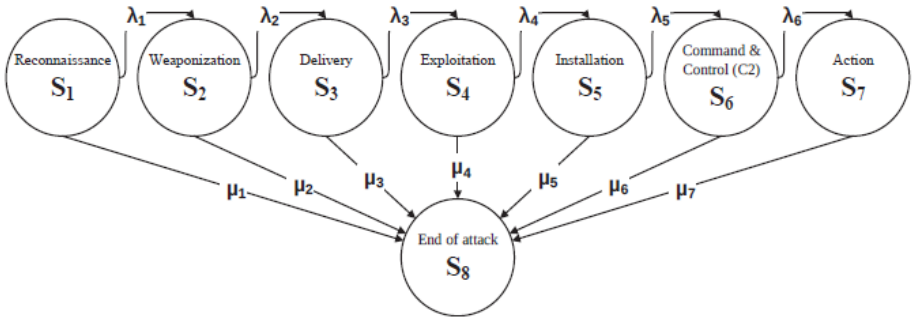


Figure 26 - LMCKC Simple State Transition Diagram (Hoffmann, 2019)

As described in the treatments above, kill chains also need to model the iterative nature of cyber-attacks as the attacker moves through the key states. One example of this is shown below

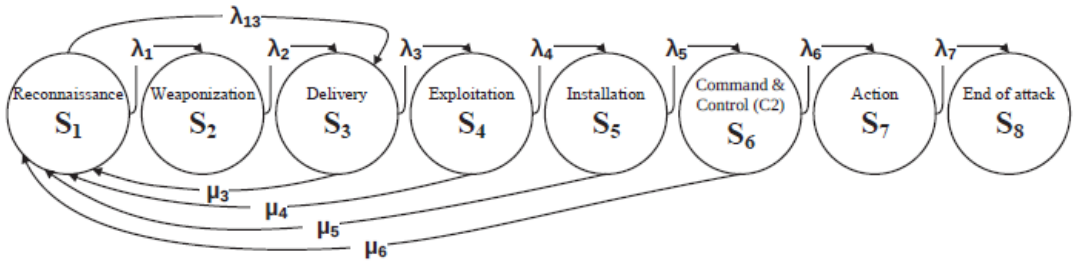


Figure 27 - LMCKC Simple State Transition Diagram (Hoffmann, 2019)

In this case the author proposes that these models may be used by cyber risk managers as input to cyber risk estimation.

Extended developments are to be found, such as in (Ioannou et al., 2019) where a specific discussion is presented on modelling the behaviours of eXfiltration Advanced Persistent Threats (XAPTs) (using a 'Markov Multi-phase Transferable Belief Model') to deal with some of the limitations presented by the Kill Chain when dealing with multi-phase / step attacks.

Subsequent work in (Caltagirone et al., 2013) introduces the Diamond Model of intrusion analysis. This paper (p52) describes how this model is complementary to the Cyber Kill Chain and provides additional analysis helping cluster attacks. Where the Kill Chain provides understanding of the attack steps the Diamond Model "allows analysts to develop tradecraft and understanding in order to build and organize the knowledge necessary to execute the Kill Chain analysis". The Diamond Model "provides an effective (but not necessarily comprehensive) list of features that should be present in every event".

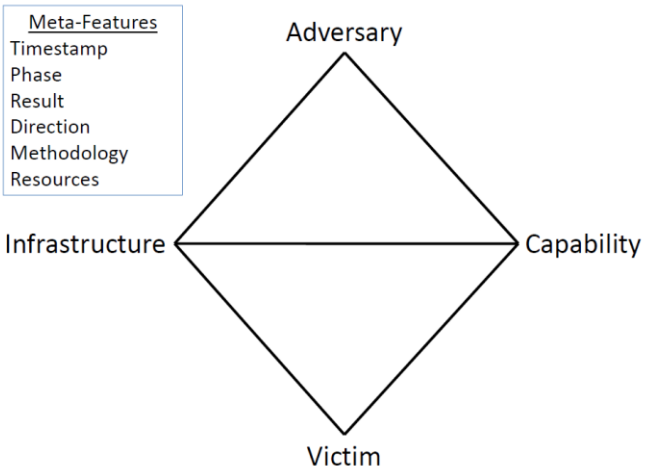


Figure 28 - Diamond Model (Caltagirone et al., 2013)

The Kill Chain concept has been developed and enriched by MITRE in their ATT&CK framework. This framework is described in more detail below, but broadly this is intended as a mid-level model to describe the actions of a malicious attacker (4.1.3 (Strom et al., 2020)). Sitting below (more detailed than) high level models such as the Cyber Kill Chain or STRIDE and above (less detailed than) exploit and vulnerability databases (and detailed indicators of compromise). ATT&CK addresses the lack of details in post compromise intrusion steps of LMCKC.

2.3.7.3 Cyber Kill Chain and Defence

The CKC refers to the Department of Defense information operations doctrine, which defines the defensive options as a detect, deny, disrupt, degrade, deceive, or destroy

Table 1: Courses of Action Matrix

Phase	Detect	Deny	Disrupt	Degrade	Deceive	Destroy
Reconnaissance	Web analytics	Firewall ACL				
Weaponization	NIDS	NIPS				
Delivery	Vigilant user	Proxy filter	In-line AV	Queuing		
Exploitation	HIDS	Patch	DEP			
Installation	HIDS	"chroot" jail	AV			
C2	NIDS	Firewall ACL	NIPS	Tarpit	DNS redirect	
Actions on Objectives	Audit log			Quality of Service	Honeypot	

(Hutchins et al., 2011)

Alternatively, a defensive approach may be formulated along the lines of NIST's Cyber Security Framework, which is more tailored to the cyber context. The options that are available to defenders could then be defined as know, prevent, detect, respond, and recover

A review of Artificial Intelligence application along the various phases of the kill chain above is provided in (Chomiak-orsa et al., 2019). In the Conclusion that the authors note that the AI solutions considered do not currently run autonomously and require expert users. This does not mean that this cannot be achieved in the longer term, but value remains in delivering improved

machine-based analysis to expert users to help decrease their workload and deal with critical issues.

2.3.7.4 Conclusion

Despite some of the limitations, critiques and developments of various kill chains noted above there remains a recognition that a lot of cyber defence remains response driven and that better understanding the 'attack trajectories' of attackers would potentially help defenders anticipate and disrupt attacks (Abstract and introduction - (Rege et al., 2017)).

For example, the relationship between the kill chain LMCKC and how the defender may use them to organise their cyber security responses to detect and respond to such attacks is usefully outlined in (DELL, 2014) (summarised in the diagram below).

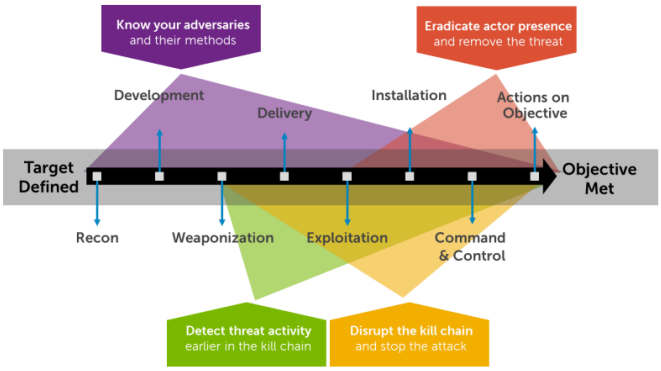


Figure 29 - Breaking The Kill Chain

The recording of these kill chains (using whatever chosen approach) then becomes additional actionable intelligence. That is, a higher-level indicator that can be used to detect the possible existence of an attack. A discussion on linking these higher level indicators with lower level indicators is presented in (Brazhuk, 2021) (building on previous work in (Brazhuk, 2019)). Specifically, this discusses linking to the "ATT&CK, CAPEC, CWE, CVE security enumerations". The ATT&CK enumerations are described in a little more detail later in this document but in summary these provide detailed views of Tactics/Techniques, technical attack patterns, software weakness categories and known specific vulnerabilities respectively.

Different security techniques bring forward different approaches to the cyber kill chain – everyone from Gartner to Lockheed Martin defines the stages slightly differently. Alternative models of the cyber kill chain combine several of the above steps into a C&C stage (command and

control, or C2) and others into an 'Actions on Objective' stage. Some combine lateral movement and privilege escalation into an exploration stage; others combine intrusion and exploitation into a 'point of entry' stage.

2.4 Cyber Threat Intelligence

2.4.1 Introduction

When understanding possible and current threats, it is fundamental that this analysis is at least partially based on experience and 'catalogued' information and knowledge. This general concept can be termed broadly as Cyber Threat Intelligence (CTI). In this section I try to summarise key concepts

Within cyber security activities, cyber threat intelligence (CTI) is associated with six major phases (direction, collection, processing, analysis, dissemination and feedback (see also Chapter 2 (Pokorno et al., 2019))) of the traditional cycle as pictured below

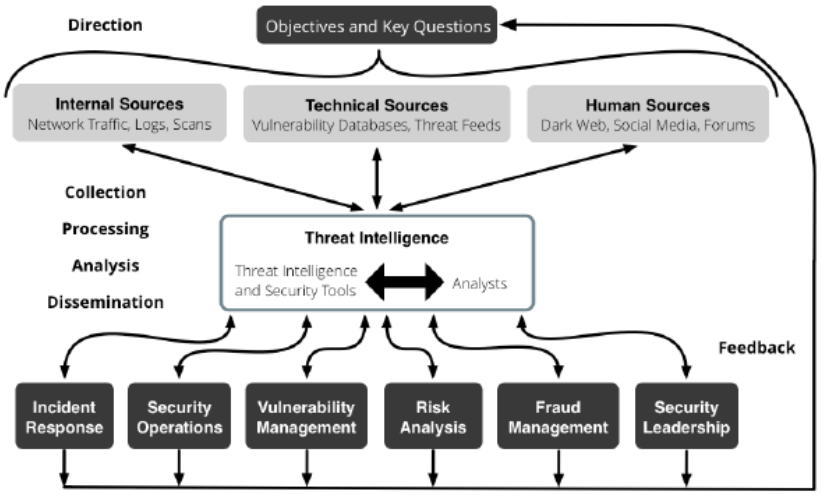


Figure 30 - Six Phases of Cyber Threat Intelligence (Pokorno et al., 2019)

Moreover, CTI uses standards, which will be briefly described below, that serve as a reference for modelling and analysing cyber threats.

2.4.2 Data, Information, and Intelligence

In 2016 through to 2019 the European Commission researched approaches to improving business awareness of risks posed by cyber-attacks (Kaiafas, 2017). The 'PROTECTIVE framework' is presented including a discussion on intelligence sharing. In section 1.1 of the introduction the investigation notes the difficulty in exactly defining the concepts of data, information, and intelligence. However, it is useful to provide some broad description within this document for at least consistency and this investigation notes the need to distinguish between Threat Data, Threat Information and Threat Intelligence expanded below).

Building on work in (ENISA, 2014) some definitions are offered. Using these (with a little refinement) for this document

Threat Data

Data refer to the low-level data collected and generated by various monitoring processes. This data may include: IDS alerts, firewall logs (including flow data and/or full packet captures), application-level logs (e.g. server log files), and operating system-level logs.

Threat Information

Data that have undergone additional processing to provide enhanced high-level insight that may help decision makers in reaching a well-informed decision.

A survey on technical threat intelligence is provided in (Tounsi & Rais, 2018). This paper recognises that cyber security approaches need to address the dynamic evolution of complex threats. It attempts to create a more specific understanding of what exactly cyber threat information is and how this may be successfully shared. In particular in the Abstract for this paper, here we have "threat intelligence means evidence based knowledge representing threats that can inform decisions" (also in (Kaiafas, 2017) (Section 1.1) we have the need for this to be "relevant, actionable and valuable").

From (Schlette et al., 2021) we have a useful overview of the key 'dimensions' we may consider when evaluating available Cyber Threat Intelligence (CTI) "extensive research on CTI has defined

its essential building blocks to comprise the threat information itself, data formats, sharing and collaboration via dedicated platforms, as well as incident response, all embraced by the topic of data quality."

From (Bromiley, 2016) we have (via R. McMillan, May 2013) "evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard." (see also (Gschwandtner et al., 2018) and (R. Brown, 2019) for similar discussions).

So based broadly on these statements, I will define Threat Intelligence (this is used interchangeably with Cyber Threat Intelligence) as:

Threat Intelligence

Evidence-based knowledge, including context, mechanisms, indicators, implications, and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decisions regarding the subject's response to that menace or hazard

In practice raw data is rarely of use without preparation, so just information and intelligence are good enough to distinguish between in general discussion.

Some of the problems faced trying to model cyber threat intelligence data are described in the introduction of (Mavroeidis & Bromander, 2017). In summary:

- Vaguely defined terminology leads to confusion amongst modellers
- Lack of formalized structuring of data leads to fields with large amounts of unstructured text
- Lack of coherent relationships between the different layers of abstraction in models

2.4.3 Types of Cyber Threat Information and Intelligence

It is worth briefly considering the types of Cyber Threat Information that is available for a Cyber Actor to use.

Tounsi in (Tounsi & Rais, 2018) discusses the division of threat intelligence into four categories:

- Strategic
    - "High level information to …. help strategists understand current risks and identify further risks of which they are yet unaware"
- Operational

    - "Information about specific impending attacks against an organisation"
- Tactical
    - "[O]ften referred to as Tactics, Techniques and Procedures and is information about how threat actors are conducting attacks"
- Technical
    - "Information normally consumed through technical resources…. typically feeds the investigative or monitoring functions of an organisation e.g., firewalls ", Intrusion Detection Systems (IDS), Indicators of Compromise (IOC) etc

In this work I will be focussing primarily on Tactical and Technical sources.

CTI is available both:

- Internally
    - Internal organisation tactical experience (recorded/machine usable and also human)
    - Event streams from various monitoring 'sensors' (e.g., Firewalls, (SIEM), IDS packages)
- Externally
    - Collated and structured intelligence sources at various levels

External CTI can be found via either commercial or openly available sources.

2.4.4 The Case for Intelligence Sharing

There has been much discussion around the benefits or otherwise of sharing CTI, but the benefits case for the sharing of (appropriate) intelligence has become generally accepted.

The approaches, benefits and barriers of sharing CTI are investigated in a good number of papers (e.g. (Skopik et al., 2016) (Rizov, 2018) (Zibak & Simpson, 2019a) (Nicholas, 2017) (Pedrinaci & Domingue, 2010) (Koepke, 2017) (Abu et al., 2018) (European Union, 2015)). Whilst the conclusion across these papers is that, broadly this is beneficial, concerns remain around preserving appropriate privacy and the ability of actors to process this data and gain maximum value.

In (R. Brown & Lee, 2019) CTI usage is reviewed and finds that a large proportion of businesses are using shared CTI (consideration of how this is related to threat management is discussed in (S. Brown et al., 2015)).

Many papers also discuss the generic concept of a Threat Intelligence Management Platform (TIMP) (e.g. (Dandurand & Serrano, 2013)). Generalised models describing how Threat Intelligence (from multiple sources) is managed into and within the organisation. In (Zibak et al., 2021) the authors investigate the factor that contribute to the success (or otherwise) of such platforms. Through this they attempt to provide a more rigorous framework to measure effectiveness.

2.4.4.1 Approaches & Standards

It is important that having gathered and structured CTI, approaches to sharing that information are also agreed (Kampanakis, 2014). (Zibak & Simpson, 2019b) and (Sauerwein et al., 2017) provide examples of the much larger discussion relating to the complex issue of standardising sharing approaches

A number of data structuring and sharing approaches are discussed in (Asgarli & Burger, 2016) (Luiijf et al., 2017) such as

- Structured Threat Information Expression (**STIX**) (OASIS, 2017)
- The transport standard Trusted Automated eXchange of Indicator Information (**TAXII**) (OASIS, 2018)
- Incident Object Description Exchange Format (Version 2) (**IODEF)** (Danyliw, 2016).

Amongst many others developments the Mandiant Open Indicators of Compromise (OpenIOC) Framework is discussed in (Gibb & Kerr, 2013)

An important source of intelligence provided in the NIST vulnerability dataset is presented in a Javascript Object Notation (**JSON**) (IETF, 2017) schema (NIST, 2019b). This adheres to the Security Content Automation Protocol (**SCAP**) (NIST, 2019d) (Waltermire & Fitzgerald-Mckay, 2018) providing a broad confederacy of interoperable security automation standards.

A common format and important for sharing intelligence is found in the STIX (and SCAP) format (mentioned above). A broad overview is shown in the figure below (Jordan, 2016):

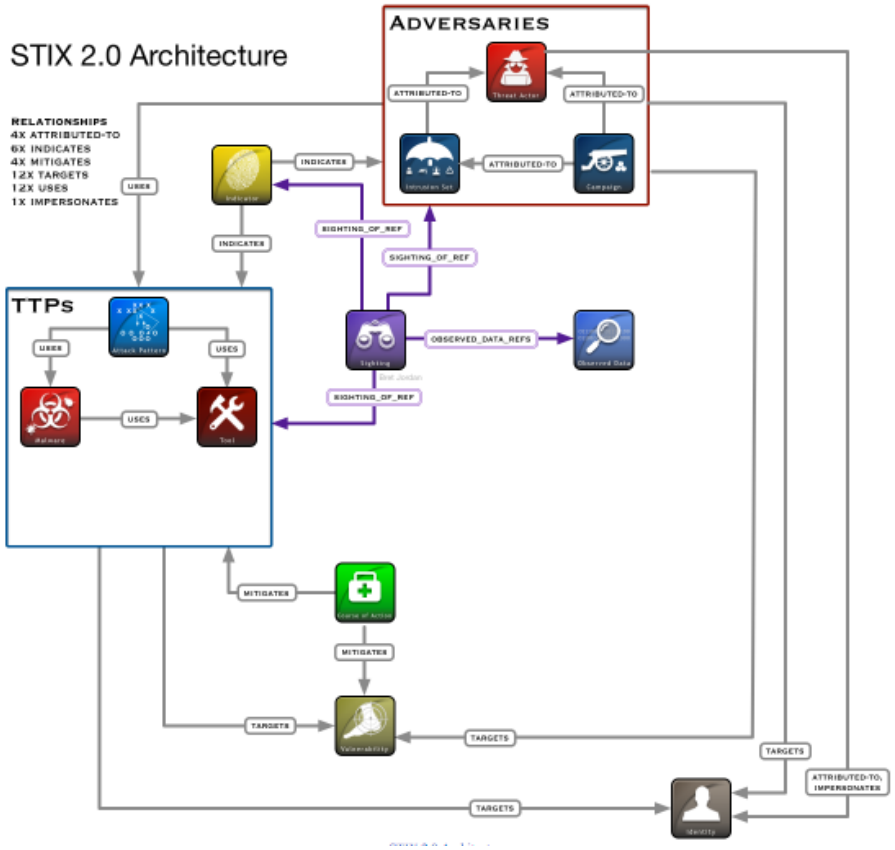


Figure 31 - STIX 2.0 Architecture

Standards also exist for alert event intelligence structuring. Examples include the Intrusion Detection Extensible Alert (IDEA) (E-infrastruktura CESNET, 2020) and Intrusion Detection Message Exchange Format (IDMEF) (Debar et al., 2007). Also, Sigma providing an open, flexible format for describing log events “The main purpose of this project is to provide a structured form in which researchers or analysts can describe their once developed detection methods and make them shareable with others.” (Sigma, 2022)

Another important open threat sharing initiative is the Malware Information Sharing Platform (MISP) (MISP, 2018b) with associated exchange format (CIRCL, 2020). An openly accessible vulnerability and exploit sharing platform is briefly described below.

Also of interest is the Vocabulary for Event Recording and Incident Sharing (VERIS) (Verizon, 2022). This is intended to provide a common standard for documenting security incidents. This also includes the VERIS Common Attack Framework which is intended to integrate VERIS with MITRE ATT&CK. This is an example of the way MITRE ATT&CK can be extended with additional techniques descriptions.

2.4.5 General Data Quality

The issue of data quality is an important constraint to the effective use of CTI. One complexity remains understanding appropriate quality measures, when the fitness for use of the data is so closely related to the specific requirements for the processing. A comparative analysis of cyber-threat intelligence sources, formats and languages is provided in (Ramsdale et al., 2020).

Some studies into cyber threat data quality dimensions have been done (e.g. (Sillaber et al., 2016) and (Schlette et al., 2020)), but currently there is no overall consensus on the key issues. This is likely to be because of the complexity of related dimensions that contribute to the overall quality (effectiveness to user) experienced. The relationship between the need to withhold intelligence retain appropriate privacy and the overall value of the intelligence is an important area of consideration (Maschmeyer et al., 2020)).

2.4.6 Openly Available Cyber Threat Intelligence

Archives

Offensive Security – Exploit Database (OffensiveSecurity, 2019)	"The Exploit Database is a CVE compliant archive of public exploits and corresponding vulnerable software, developed for use by penetration testers and vulnerability researchers. Our aim is to serve the most comprehensive collection of exploits gathered through direct submissions, mailing lists, as well as other public sources, and present them in a freely available and easy-to-navigate database. The Exploit Database is a repository for exploits and proof-of-concepts rather than advisories, making it a valuable resource for those who need actionable data right away."
APTNotes (Bandla & Westcott, 2019)	"APTnotes is a repository of publicly-available papers and blogs (sorted by year) related to malicious campaigns/activity/software that have been associated with vendor-defined APT (Advanced Persistent Threat) groups and/or tool-sets"
Carnegie Mellon University Vulnerability Notes Database (University, 2019)	"The Vulnerability Notes Database provides information about software vulnerabilities. Vulnerability notes include summaries, technical details, remediation information, and lists of affected vendors. Most vulnerability notes are the result of private coordination and disclosure efforts. For more comprehensive coverage of public vulnerability reports, consider the National Vulnerability Database (NVD). CERT/CC also publishes the Vulnerability Notes Data Archive on GitHub"

Feeds

NCSC Threat Reports (NCSC, 2019)	Openly accessible Threat Reports derived from other open-source reporting.

FireEye (public threat reports) (FireEye, 2019)	"FireEye posts blog entries under threat research to present and discuss cyber-attacks and threat intelligence from a technical perspective. These blog posts cover everything from exploits and vulnerabilities to advanced malware and targeted attacks."
IBM XForce Exchange (IBM, 2019)	"IBM® X-Force Exchange is a cloud-based, threat intelligence sharing platform that you can use to rapidly research the latest global security threats, aggregate actionable intelligence, consult with experts and collaborate with peers. IBM X-Force Exchange, supported by human- and machine-generated intelligence, leverages the scale of IBM X-Force to help users stay ahead of emerging threats."
Computer Incident Response Centre Luxembourg (CIRCL) (CIRCL, 2019)	"A government-driven initiative designed to gather, review, report and respond to computer security threats and incidents". An openly accessible source of CTI (vulnerabilities and recorded exploits), it is also worth noting this is implemented alongside the open source threat sharing and management platform Malware Information Sharing Platform (**MISP**) MISP (CIRCL, 2019) (Wagner et al., 2016) (MISP, 2019) (MISP, 2018a).
AlienVault Open Threat Exchange (OTX) (AlienVault, 2019b)	"The world's largest open threat intelligence community that enable collaborative defense with actionable, community-powered threat data". Now owned by AT&T, it is also worth noting that this integrates with the open source SIEM tool Open Source Security Information Management (**OSSIM**) (AlienVault, 2019a).
National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD) (NIST, 2019c)	"The NVD is the U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics." This includes Common Vulnerabilities and Exposures (**CVE**) and the Common Weakness Enumeration (CWE).
Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) (MITRE, 2019c) (Bodeau et al., 2018b)	"The MITRE ATT&CK knowledgebase describes cyber adversary behaviour and provides a common taxonomy for both offense and defense. It has become a useful tool across many cyber security disciplines to convey threat intelligence, perform testing through red teaming or adversary emulation, and improve network and system defenses against intrusions." ATT&CK (which continues to develop and is updated quarterly) describes patterns of behaviour in terms of (Strom et al., 2017): • *Tactics* – Highest level of abstraction and represent tactical goals of an adversary. • *Techniques* – Actions taken by adversaries to achieve the tactical goals

- *Procedures* – Specific ways in which an adversary implements a technique

There are three major groupings within ATT&CK Pre-ATT&CK, Enterprise and Mobile. Representing patterns in these major domains.

- *Enterprise ATT&CK* defines Tactics that can be seen as an expansion to the latter part (post Weaponize) of the Cyber Kill Chain.
- *Pre-ATT&CK* defines Tactics that can be seen as an expansion to the Reconnaissance and Weaponize elements of the Cyber Kill Chain.
- *ATT&CK mobile* defines Tactics used by Malicious Attackers in a mobile environment.

Common Attack Pattern Enumeration & Classification (CAPEC) (MITRE, 2019a)

"Understanding how the adversary operates is essential to effective cyber security. CAPEC™ helps by providing a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. It can be used by analysts, developers, testers, and educators to advance community understanding and enhance defenses."

The **Attack Patterns** (attributes and approaches of different common Attacks) are organised by Mechanism and Domains of the Attacks.

Malware Attribute Enumeration & Characterisation (MAEC) (Kirillov et al., 2015)

"Malware Attribute Enumeration and Characterization (MAEC) is a standardized language and format being formulated in cooperation with industry, government, and academia for use in attribute-based malware characterization. MAEC is composed of a set of attribute enumerations, a schema, and a standard output format for the transport and communication of MAEC-encoded data. MAEC is being developed by MITRE under the sponsorship of DHS NCSD and others and will be part of MITRE's Making Security Measurable (MSM) effort."

The Malware characterisation is closely integrated with relevant CVE, CWE and CPE entries (see also below).

Common Weakness Enumeration (CWE) (MITRE, n.d.-b)

".. A community-developed list of common software security weaknesses. It serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts."

The vulnerability descriptions (may) link to standard weakness descriptions that provide categorisation. The CWE is described through a hierarchy of levels:

- *Category* – Highest level of categorisation
- *Class* – "A weakness that is described in a very abstract fashion, typically independent of any specific language or technology"

	• *Base* – "A weakness that is described in an abstract fashion, but with sufficient details to infer specific methods for detection and prevention" • *Variant* – "A weakness that is described at a very low level of detail, typically limited to a specific language or technology"
Verizon DIBR Report 2022 (Verizon, n.d.) ***Additional sources***	Added here as an example of detailed information about various breaches available across various platforms.
Snort (Roesch, 2019)	Free open-source software (now developed by Cisco). Snort has three modes of operation Sniffer, Packet Logger and Network Intrusion Detection System (**NIDS**) mode. In the NIDS mode the software uses intelligence provided through a team of analysts ((Cisco, n.d.)) to provide additional information linking network alerts to known vulnerabilities (e.g. CVEs). The rules used to achieve this are regularly updated and can be downloaded.
ThaiCert (ThaiCert, 2023)	"This portal aims to create full profiles of all threat groups worldwide that have been identified with all research generously shared by anti-virus and security research organizations over the years. It can be used as "threat group cards", as the portal title suggests, to have everything together in an elaborate profile for each threat group. All dates shown in the cards are the dates when the stated activities started, not necessarily when the reports about them came out. All information in this portal comes from public sources (OSINT). The difficult part of attributing campaigns to actors has been done by those security research organizations as well. What makes this difficult is the fact that there may be some overlap between threat groups, where they share tools or people move between groups, or when groups suddenly change tactics or type of target."

Table 2 - Openly Available Cyber Threat Intelligence

There are also studies on how to extract Open Source (Cyber Threat) Intelligence (OSINT) from various sources. An example of this is given in (Tundis et al., 2022) with a good review of related literature.

2.4.7 Connecting Intelligence

MITRE / NIST ATT&CK, CAPEC, CWE and CVE are freestanding intelligence sources. There exists some limited basic interoperability between these datasets. Vulnerabilities descriptions in CVE

normally include information about related Weaknesses (CWE). Attack Patterns (CAPEC) include information about the Weaknesses exploited. Techniques that may be used by malicious agents are sometimes (but rarely) linked to underlying Attack Patterns. However, the linkage across these sets remains quite sparse and often the precise relevance depends on attack context (provided through textual reports or manual intervention of analysts).

Some investigation into linking models is discussed in (Brazhuk, 2019) (Brazhuk, 2021) and (Brazhuk, 2022) where the author investigates semantic and language modelling across CAPEC and CWE. In (Hemberg et al., 2020) the author seeks to ATT&CK, CWE, CVE, and CAPEC to assist threat hunters. The linkages are expressed through a graph model (BRON). The authors note the limitations on what can be achieved due to the underlying quality of the public data. In (Kurniawan et al., 2021) the authors consider a semantic expression of ATT&CK in RDF-S and OWL to provide for greater semantic interoperability. In (Kuppa et al., 2021) and (Grigorescu et al., 2022) the authors discuss specifically linking CVE to ATT&CK Techniques and this also exists as a MITRE initiative (MITRE, 2022e).

2.4.8 Conclusion

Key intelligence OSINT data such as those provided MITRE and NIST are used across many sites. They are commonly used alongside software packages (e.g., IDS and SIEM) that help cyber-analysts to detect and mitigate suspicious activity.

This intelligence covers a wide scope but true interoperability across the piece is still an area of research and development. There is a wide range of open-source textual descriptions of APT attacks and campaigns. These textual reports are often provided by experts and are also used as source material for more general reporting across the press. However, these reports are not machine readable. Some Natural Language Processing (NLP) based approaches are discussed below but they currently still provide limited effectiveness.

Specifically, there appears to be a lack of openly-available records of known cyber-attacks expressed as a sequence of ATT&CK techniques used.

2.5 Cyber Situational Awareness

Cyber Situational Awareness See also intro (and section 2.4) to “A Predictive Framework for Cyber Security Analytics Using Attack Graphs”

Three/Four levels of Cyber SA

- Perception
- Comprehension
- Projection
- Resolution

Attack intention recognition is a part of the primary objectives of cyber situation comprehension and our work is mainly focused on the attack intention recognition of APT

2.5.1 Introduction

2.5.2 Situational Awareness

A good overview of general Situational Awareness (and its application to safety concepts) is given in (Stanton et al., 2001). Section 1 provide a background to the history of the concept noting the initial use of the term in World War 1 and subsequent research and development within the aviation industries in response to the complexities for pilots and air traffic controllers. “This call has arisen with the increased realisation that system design is no longer optimised for human operation and, under some conditions, has ‘overstepped the human’s capability to keep track.’”. Three overarching theories of situational awareness are also outlined and discussed.

From Section 2 we have a very summary:

Situation Awareness

“At a very simple level, situational awareness is an appropriate awareness of a situation” (my underlining) (in turn derived from (Smith & Hancock, 1995))

Similarly from (Gawron, 2019) (opening paragraph) we have “Situational Awareness (SA) is knowledge relevant to the task being performed. For example, pilots must know the state of their aircraft, the environment through which they are flying, and relationships between them, such as thunderstorms are associated with turbulence.”

Together these provide a flavour of the highly contextual nature of situational awareness (and this remains true in Cyber Situational Awareness discussed later below).

2.5.3 Situational Awareness Models

A number of Situational Awareness framework definitions/models have been developed.

For example from (Stanton et al., 2001) (Section 2) we have a discussion on three major approaches

- Endsley (Endsley, 1988)
    - "Situational awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and a projection of their status in the near future"
- Bedny & Meister (Bedny & Mesiter, 1999)
    - "Situational awareness is the conscious dynamic reflection on the situation by an individual. It provides dynamic orientation to the situation, the opportunity to reflect not only the past, present, and future, but the potential features of the situation. The dynamic reflection contains logical-conceptual, imaginative, conscious, and unconscious components which enables individuals to develop mental models of external events"
- Smith & Hancock (Smith & Hancock, 1995)
    - "Situational awareness is the invariant in the agent-environment system that generates the momentary knowledge and behaviour required to attain the goals specified by an arbiter of performance in the environment"

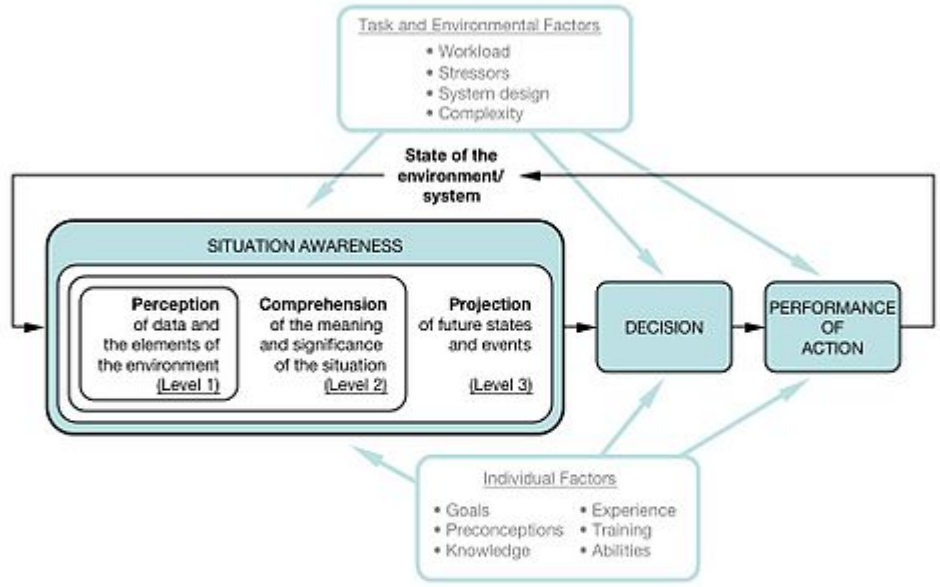
These three approaches are contrasted and compared and shown to be understandable in terms of a common model where an agent with 'knowledge and mental models' can compare these to perceived information (from the world around them) and 'reflect and project' toward a response. The approaches described above can be seen to address such a model but with differing emphases on the various elements.

Further models are discussed in (Raulerson, 2013) where we also have illustrations of models from (White, 1991) and (Steinberg et al., 1999). However, the general point above applies.

2.5.4 Cyber Situational Awareness Models

Cyber Situational Awareness (CSA) is considered an application of situational awareness to Cyber Space (Franke & Brynielsson, 2014) (Brynielsson et al., 2016). A snapshot of related theory and models is provided in (Liu et al., 2017). The study of an overall CSA has led to a number of different approaches, however Endsley's situation reference model (Endsley, 1995b) (Endsley, 1995a) (a copy of the McGuinness and Foy paper has been difficult to find but an overview is given in (Gawron, 2019) 3.2.2) forms a seminal basis to much traditional thinking around CSA (c.f. Introduction from (Franke & Brynielsson, 2014) and section 2.1 of (Jajodia et al., 2010)).

This is illustrated below



Endsley's model of SA. This is a synthesis of versions she has given in several sources, notably Endsley (1995a) and Endsley et al (2000). Drawn by Dr. Peter Lankton, May 2007.

Figure 32 - Endsley's Model Of Situation Awareness (Wikipedia, 2022)

An early ontology for Situational Awareness (SAW) was developed in (Matheus et al., 2003). This creates a framework for data fusion extending the Joint Directors of Laboratories (JDL) Data Fusion model (shown below) (Steinberg et al., 1999)

Fusion Level	Association Process	Estimation	Entity Estimation
L.0 Sub-Object Assessment L.1 Object Assessment	Assignment	Detection Attribution	Signal Physical Object
L.2 Situation Assessment L.3 Impact Assessment	Aggregation	Relation Plan Interaction	Aggregation Effect (Situation given Plans)
L.4 Process Refinement	Planning	(Control)	(Action)

Figure 33 - JDL 5 Levels Of Data Fusion (Matheus et al., 2003)

Or relating more directly to the Endsley model.

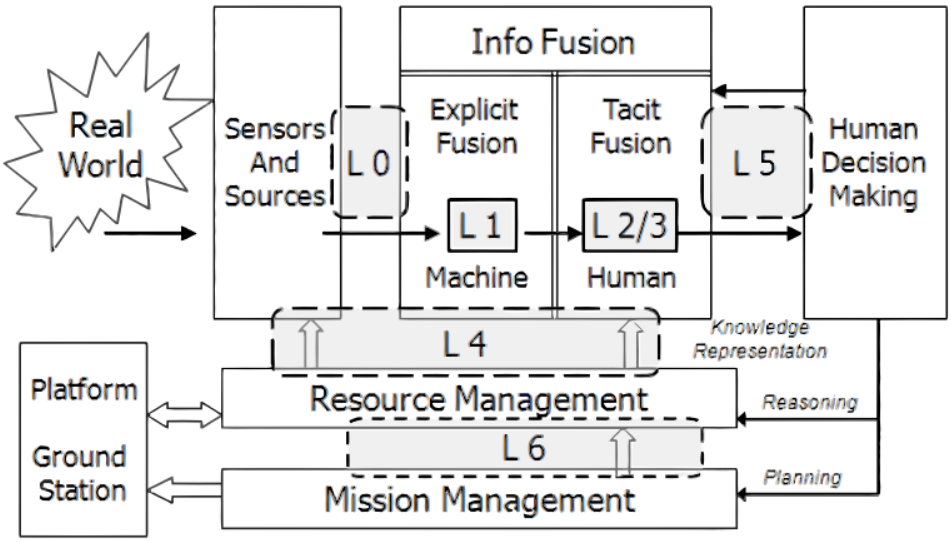


Figure 34 - Data fusion information group (DFIG) model (Han et al., 2013)

There is some criticism of the implied sequential nature and lack of human-in-the-loop of the JDL (L0-4) and DFIG (+L5) models, but they do provide a start in building a data fusion process to underpin a situational awareness model.

An ontology (model/theory of entities and the relationships between them) is then proposed, blending JDL, Endsley's Human Computer Interaction (HCI) and logic. This ontology is intended to be applicable to generic (rather than specific) situations.

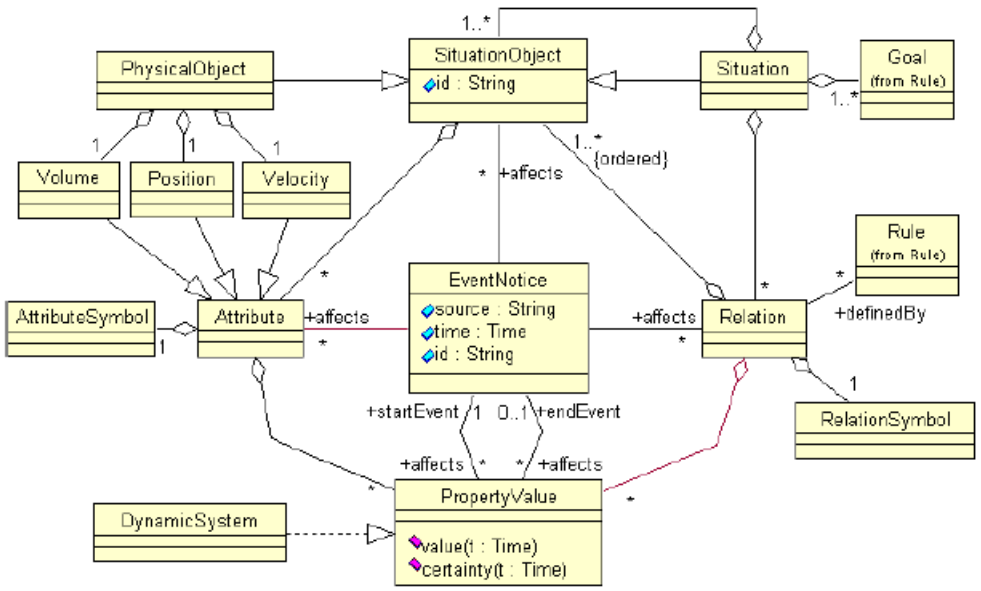


Figure 35 - Core SAW Ontology (Matheus et al., 2003)

In (Komárková et al., 2018) the CRUSOE a data model for CSA is presented. Here the authors investigate the problem of coordinating and fusing the heterogeneous data required to support situation assessment processing. The main layers of this model are shown below

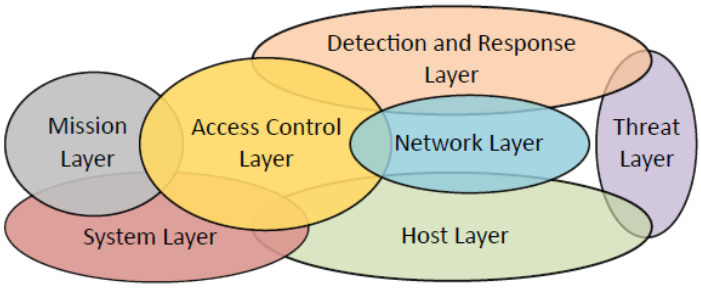


Figure 36 - CRUSOE Model Layers (Komárková et al., 2018)

These layers are then detailed further to define the model, an example from the Detection and Response Layer is shown below

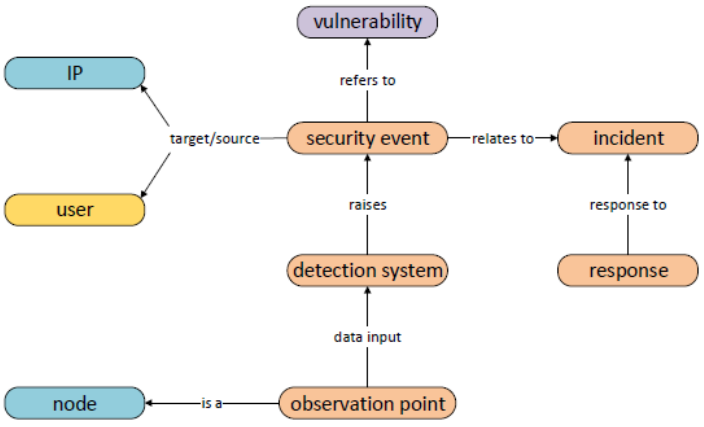


Figure 37 - CRUSOE - Detection and Response Layer (Komárková et al., 2018)

2.5.5 Outline Cyber Situational Awareness Reference Model

Based on Endsley's situation reference model and extension by McGuinness and Foy (Mcguinness, 1999) Onwubiko provides an outline reference model in (Onwubiko, 2017).

An outline of this proposed broad reference model is illustrated in (Onwubiko, 2017) and is shown below.

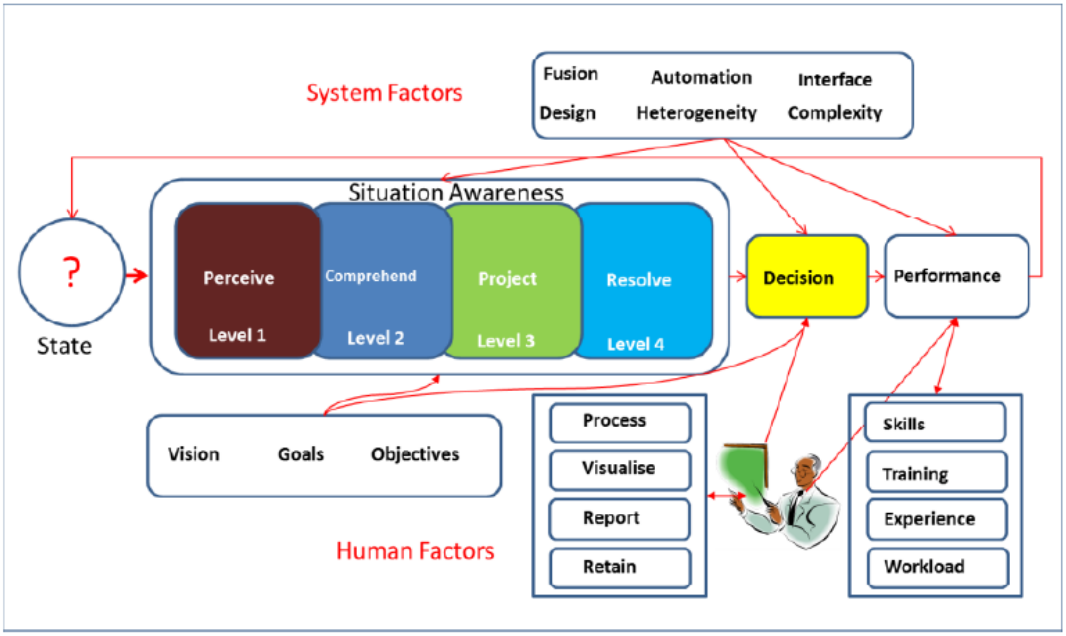


Figure 38 - Situation Awareness Reference Model (Onwubiko, 2017)

- Level 1 Perceive – Relates to the ability to 'sense' data/information within the target configuration.
- Level 2 Comprehend – Relates to the ability to analyse the data/information provided in Level 1
- Level 3 Project – Relates to the ability to use the current state and comprehension and predict next steps (tactically and strategically)
- Level 4 Resolve – Relates to specific actions to be taken deal with current issues.

This reference model is then extended to define an Instantiation Model, which provides a template allowing applications to be built 'consistently'.

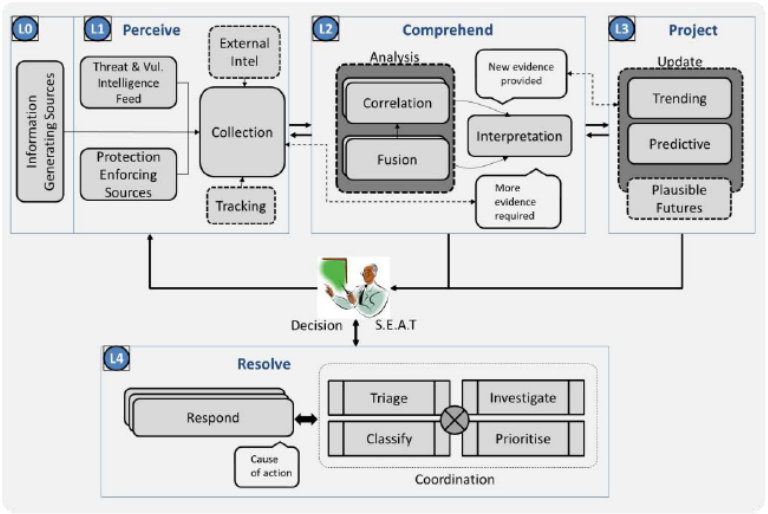


FIGURE 3: CYBER SA INSTANTIATION MODEL

Figure 39 - Cyber Situation Awareness Instantiation Model

A more detailed instance of a framework is given in (Jajodia & Albanese, 2017) and an overview figure from this paper is shown below.

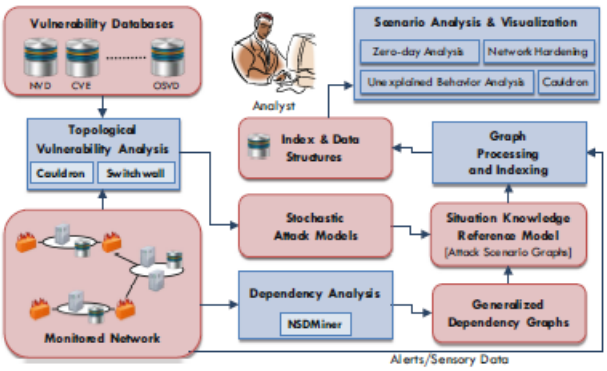


Figure 40 - Cyber Situation Awareness Framework (Jajodia & Albanese, 2017)

As noted in the Conclusion of (Jajodia & Albanese, 2017) this is broadly based around Situation Perception, Situation Comprehension and Situation Projection, so we can easily map the elements (broadly) on to the Cyber Instantiation model shown in (Onwubiko, 2017).

In (Iannacone et al., 2015) the authors propose a cyber situational awareness model for multi-phase attacks built on a 'Markov Multi-Phase Transferable Belief Model' (MM-TBM). Here the authors recognise that a multi-phase attack is built from multiple kill chains and that existing belief models did not address the need to associate data fusion approaches with the multiple 'hypothesis-spaces'.

(Alavizadeh et al., 2022) provides a survey of current state of the art cyber situation awareness systems with reference to AI based attacks. The document reviews 'key design principles, framework, classifications, data collection, and analysis of the techniques, and evaluation methods' as a precursor to future work.

Related studies have included investigations into overarching Cyber ontologies (we need to know what we are being 'aware' of!) for some time. As part of MITREs initiative in this area (Parmelee, 2010) considers an 'Ontology Architecture for Cyber-Security Standards' and a 'trade study' is documented in (Obrst et al., 2014) reviewing a 'middle-out' approach to building such a cyber ontology and providing recommendations on next steps.

In (Asgarli & Burger, 2016) the authors also consider the wider context of threat sharing (between organisations) and associated semantic ontologies, the paper examines the overlap between STIX, IODEF and OpenIOC and how RDF/OWL may provide additional benefits. In (Kang et al., 2021) a knowledge graph is developed to analyse Snort IDS alerts. This is augmented with wider public

information to provide increased situation analysis. Further in (Khairkar et al., 2013) an ontology for the detection of web attacks is presented. Here the authors extract semantic relations between computer attacks and IDS alerts and design an ontology to enable this detection.

2.5.6 Conclusion

Cyber Situational Awareness is a very wide term. Achieving insight into required elements of this is closely related to general data fusion (Iannacone et al., 2015) and interoperability problems. Over and above full automation of solutions, the support and easing of the cyber-analyst's workload is valuable. Here, visualisation of relevant situational status measures is also important (e.g. (Jiang et al., 2022) and (Franklin et al., 2017)).

MITRE and how their various initiatives relate to Cyber Situational Awareness is described (Noel & Heinbockel, 2015). Of note here is ATT&CKs role with respect to threat analysis.

2.6 Conclusion

Investigations reveal that there appears to be a lack of openly-available records of known cyber-attacks expressed as a sequence of the techniques used in those attacks (see Conclusion).

MITRE ATT&CK provides a contribution to Cyber Situational Awareness and Cyber Threat Analysis (see Cyber Situational Awareness) but lacks this type of intelligence required to understand cyber-attack sequences.

Chapter 3 Related Work

3.1 Introduction

This Chapter provides background on specific areas related to the Research Questions and the approach taken. It is also used to confirm the 'gap' being investigated.

It discusses the following:

- Automatic ATT&CK Intelligence Extraction from Attack Reports (see Automatic ATT&CK Intelligence Extraction from Attack Reports).
- Attack Classification (see Attack Classification).
- Approaches to Automatic Kill Chain Detection (see Approaches to Automatic Kill Chain Detection).
- Attack Modelling Languages (see Attack Modelling Languages).
- Sequence Comparison (see Sequence Comparison).
- Markov Models (see Markov Models)

It also describes the challenge to be addressed in this Thesis (see The Challenge)

3.2 Automatic ATT&CK Intelligence Extraction from Attack Reports

3.2.1 Introduction

Some work was undertaken to attempt automation of the creation of test attack sequences from the relevant attack reports. This proved to be a major undertaking and the outputs were of dubious 'precision'.

This section is included to outline some related studies. It is also added to clarify the decision to continue with the more labour-intensive manual creation of the sequences.

3.2.2 Review

Although some investigation and experimentation with some of the approaches was pursued while researching for this work, it became clear that this would be a major piece of work. To that end I have noted that this would be an interesting (and important) line of investigation to pursue in the Future Work section below.

A general background is surveyed in (Rahman et al., 2020). In this work the authors review 38 studies and synthesise the purposes. Although much work is available this focusses primarily on identifying 'point' information (e.g., IOCs and specific TTPs) rather than sequences of actions. This report also distinguishes between direct mining and Natural Language Processing (NLP) type approaches. Several sources were considered including Threat Reports, Twitter Feeds, Forums, Web Logs, Version Controlled Repositories, System and Application Logs and the Darknet.

Two papers (Noor et al., 2019) and (Noor et al., 2018) consider application of NLP to threat attribution (specifically FinTech) and a CTI and Association Rule Mining and machine learning framework to identify most prevalent TTPs and relevant association rules.

Building on previous work in (Z. Zhu & Dumitras, 2018) to extract the semantics of malicious

campaigns form threat intelligence reports, a specific tool (rcATT) to automatically extract ATT&CK TTPs from reports is developed in (Legoy et al., 2020) (see also (Lin et al., 2021) below). MITRE themselves have made available their own attempt at a tool (Threat Report ATT&CK Mapper (TRAM)) in (MITRE, 2019d). Both provide access to the source code and when the code was downloaded and adjusted produced basic results from test runs on several sample reports. They do provide a platform for further development although they are based around the standard term frequency-inverse document frequency (TF-IDF) approach.

Several papers review language patterns that may be used to automate analysis. In (Andrei Brazhuk, 2019) and (Brazhuk, 2022) the authors investigate natural language phrases that may be used to identify attack patterns (e.g. CAPEC and CWE see also (Kanakogi et al., 2022)). In (Niakanlahiji et al., 2019) the authors develop SECCMiner using NLP to extract tactics and techniques from textual reports using this to conduct a trend analysis of use over a period of several years. Related papers include (Tundis et al., 2022) that investigates building measures that can be used to understand the quality of CTI before processing and (Z. Yu et al., 2022) where a Convolutional Neural Network approach is used to classify tactics and techniques in CTI.

In (Husari et al., 2017) the authors develop TTPDrill to extract threat actions from textual reports. The authors note the difficulty of analysing these reports due to the lack of 'standard languages and automated analytics'. Here they seek to develop analytics to learn TTPs and link them to Kill Chain phases, as well as making this available in a standardised STIX form. In a subsequent poster (Husari et al., 2019) the authors consider going further to learn chains of actions from cyber threat intelligence and associate with APTs. At the time of writing, it is not obvious that this has yet been developed further.

The 'Extractor' tool developed in (Satvat et al., 2021) also attempts to extract attack behaviour from reports. This tool uses an NLP approach ("Semantic Role Labeling (SRL), a processing model that can detect semantic relationships among entities in a sentence") to extract the required information and present the relevant attack details in the form of a graph.

3.2.3 Conclusion

The issue of extracting structured intelligence from unstructured reports has attracted several lines of investigation including sophisticated NLP based approaches. This need is arising due to the lack of 'standard languages and automated analytics' (Husari et al., 2017). Attempts to use a couple of the relatively straightforward proposals yielded some extraction of intelligence but this was incomplete and certainly did not provide the required sequences. For this reason, a manual approach was taken to extracting the required intelligence to demonstrate an associated structuring of this temporal sequencing.

3.3 Attack Classification

3.3.1 Introduction

This section reviews existing work on cyber-attack classification models. A synthesis of these various proposals will be subsequently used as the basis of a meta data model for a collection of cyber-attacks.

The works discussed here have been found through keyword search in Google Scholar and then following references and related citations.

3.3.2 Cyber Attack Classification Models

There are several proposals on this specific topic but some of the key proposals are outlined here.

A useful survey provided in "A survey on various cyber-attacks and their classification" (Uma & Padmavathi, 2013). This paper notes the need to understand cyber-attacks and their classification to guide defensive planning. It discusses common high-level approaches used in the classification of cyber-attacks. These are itemised as below:

- Based on Purpose
    - Reconnaissance Attack, Access Attack and Denial of service Attack
- Legal Classification
    - Cyber-crime, Cyber espionage, Cyber terrorism and Cyberwar

- Based on severity of Involvement
    - Active Attacks and Passive Attacks
- Based on Scope
    - Malicious Large Scale and Non-Malicious Small Scale
- Based on Network Types
    - E.g. Mobile, Adhoc Networks and Wireless Sensor Networks

This leads to a high-level classification model (in the form of a list) of Reconnaissance, Access, Denial of Service attacks, Cyber-crime attacks, Cyber espionage attacks, Cyber terrorism attacks, Cyberwar attacks, Active attacks, Passive Malicious attacks, Non-Malicious, Attacks in MANET (Mobile Ad hoc Networks), Attacks on WSN (Wireless Sensor Networks).

An influential publication 'A taxonomy of network and computer attacks' (Hansman & Hunt, 2005) considers previous taxonomies (e.g. above) and approaches to creating such a taxonomy. Building on a critique of previous proposals the paper then proposes a new taxonomy and demonstrates its use against fifteen attacks. This provides a good starting point as it provides a good foundation for key principles.

The paper begins by outlining requirements for a 'good' taxonomy. In summary these are stated as follows:

- **Accepted**: The taxonomy should be structured so that it can become generally approved.
- **Comprehensible**: A comprehensible taxonomy will be able to be understood by those who are in the security field, as well as those who only have an interest in it.
- **Completeness**: For a taxonomy to be complete/exhaustive, it should account for all possible attacks and provide categories accordingly. While it is hard to prove a taxonomy that is complete or exhaustive, it can be justified through the successful categorisation of actual attacks.
- **Determinism**: The procedure of classifying must be clearly defined.
- **Mutually exclusive**: A mutually exclusive taxonomy will categorise each attack into, at most, one category.
- **Repeatable**: Classifications should be repeatable.
- **Terminology complying with established security terminology**: Existing terminology should be used in the taxonomy so as to avoid confusion and to build on previous knowledge.
- **Terms well defined**: There should be no confusion as to what a term means.
- **Unambiguous**: Each category of the taxonomy must be clearly defined so that there is no ambiguity with respect to an attack's classification.
- **Useful**: A useful taxonomy will be able to be used in the security industry and particularly by incident response teams.

The paper also notes that although these are all useful properties, but they are not necessarily mandatory for every successful taxonomy.

The paper argues that a Tree like taxonomy would be too disparate and that a simple List like taxonomy would not be that useful. It therefore proposes a straightforward taxonomy with four major dimensions:

- Attack Vector
    - "The attack vector is the method by which an attack reaches its target".
- Classification of Target
    - In this paper the target is defined as the technical component (or class of component) being attacked.
- Vulnerabilities Attacked
    - Vulnerabilities and exploits being used.
- Payloads
    - Over and above the first dimension this describes additional payload that may be introduced to potentially launch further phases/steps of attack.

The possibility of augmenting with additional dimensions if additional information is provided is also briefly explored. Demonstrating that generic classification models will be tailored to specific applications.

The CAPEC taxonomy is discussed in (Barnum, 2008). Although created some time ago it remains maintained and provides "a comprehensive dictionary of known patterns of attack employed by adversaries to exploit known weaknesses in cyber-enabled capabilities". This provides a classification model for elements of an attack.

In (Simmons et al., 2009) the authors propose a taxonomy for cyber-attacks. This is called AVOIDIT (Attack Vector, Operational Impact, Defense, Information Impact, and Target). Building on (Hansman & Hunt, 2005) this proposes five major dimensions to classify an attack. These are:

- Attack vector
- Attack target
- Operational impact
- Informational impact
- Defence

The last dimension is included to guide defenders to appropriate mitigations. This taxonomy addresses the issue of the classification of blended attacks. A blended attack is one that exploits

multiple vulnerabilities. Here the authors propose a tree-like structure to address this issue. The full taxonomy is given below

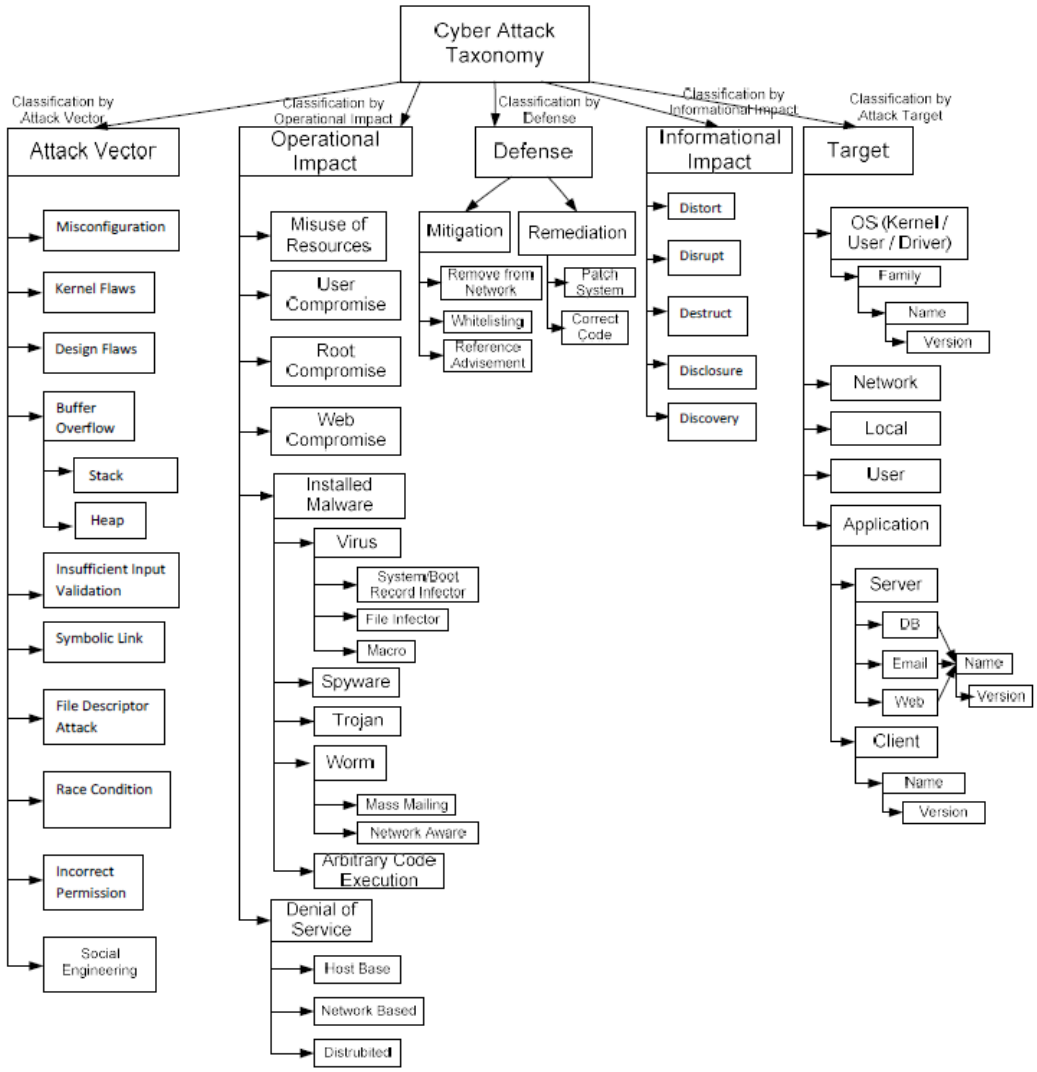


Figure 41 - AVOIDIT Attack Taxonomy (Simmons et al., 2009)

The authors then present a comparison with their taxonomy and previous taxonomies to demonstrate the additional contribution provided by this new taxonomy. The blended attack description is achieved by listing the multiple vulnerabilities exploited. The authors argue that this provides the defender with more detailed information to organise their defences. In (Bodeau et al., 2018a) Section 3.1.5 we have a small overview of well-known cyber-attack taxonomies at that time. The section is very brief and simply outlines well known taxonomies at that point (specifically noting AVOIDIT described above).

In (Meyers et al., 2009) the authors present a taxonomy of types of cyber attackers alongside a taxonomy of the attack approaches used this taxonomy of approaches is based heavily on

(Hansman & Hunt, 2005). Here the taxonomy is based on the Attack Vector (as defined above). Categorised by Types (attack class) and more detailed Subtypes. This is illustrated below

Attack Class	Subtypes	Description
viruses	file infectors, system/boot record infectors, macros	self-replicating program that replicates through infected files; attached to an existing program
worms	mass mailing via botnets, network aware	self-replicating program that replicates through networks or email; no user interaction required
trojans	remote access, data destruction	program made to appear benign that serves a malicious purpose
buffer overflows	stack-based overflows, heap-based overflows	process that gains control or crashes another process via buffer overflowing
denial of service	host (resource hogs, crashers), network (TCP, UDP, ICMP flooding), distributed	attack that prevents legitimate users from accessing a host or network
network **attacks**	spoofing, web/email phishing, session hijacking, wireless WEP cracking, web application **attacks**	attack based on manipulating network protocols, against users or networks
physical **attacks**	basic, energy weapon (HERF gun, EMP/T bomb, LERF), Van Eck	**attacks** based on damaging the physical components of a network or computer
password **attacks**/ user compromise	guessing (brute force, dictionary **attacks**), exploiting implementation	**attacks** aimed at acquiring a password or login credential
information gathering	packet sniffing, host mapping, security scanning, port scanning, OS fingerprinting	**attacks** in which no damage is carried out, but information is gathered by attacker

Figure 42 - Attack Classification Model (Meyers et al., 2009)

In (Chapman et al., 2011) the authors propose a taxonomy of cyber-attacks based on the level of access the attacker needs to access the target system. They define three major tiers:

- Tier 1 – No Network or Computer Access
- Tier 2 - User Access with Limited Privileges
- Tier 3 - Root Access/Administrative Privileges

They then allocate different major attack approaches to the relevant tiers. Following this they develop an outline of an approach to simulating cyber-attacks based on this taxonomy.

There are also several proposals on taxonomies aimed at describing specific classes of attack.

An example of this in (B. Zhu et al., 2011) proposes a taxonomy of attacks aimed against SCADA systems. There has (understandably) been an increased interest in understanding the vulnerabilities that exist in Industrial Control Systems (ICS). ATT&CK now includes a specific matrix dedicated to techniques used in these types of attacks (c.f. (MITRE, 2022d) and used in (Toker et al., 2021)). In (S. Kim et al., 2019) we also see a “Cyber-attack taxonomy for digital environment in nuclear power plants” this is illustrated below

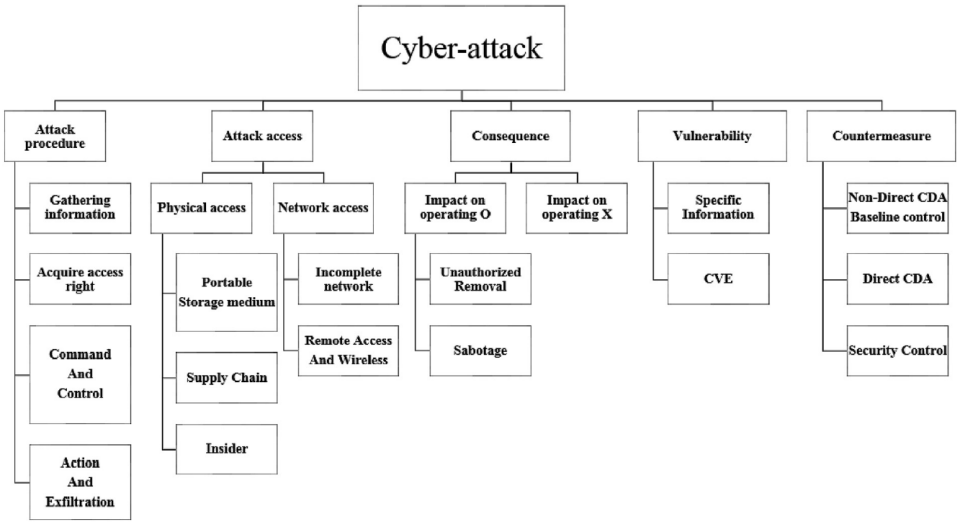


Figure 43 - Cyber-attack taxonomy for dig. Env. in nuclear power plants (S. Kim et al., 2019)

Although specific to an environment, we can still note the major dimensions of

- Attack Procedure
- Attack Access
- Consequence
- Vulnerability
- Countermeasure

In (Pöhn & Hommel, 2022) the authors propose “TaxIdMA: Towards a Taxonomy for Attacks related to Identities”. This taxonomy is intended to support classification of attacks associated with identity management systems. It is evaluated against reports on eight real-world attacks. It the proposes an ‘Attack Background’ taxonomy which is illustrated below. The four main ‘dimensions’ are

- Attacker (based on (Chng et al., 2022))
- Target (based on (Hansman & Hunt, 2005) and (Simmons et al., 2009))
- Identity (based on (Chapman et al., 2011))
- Attack (based on (Hansman & Hunt, 2005) and (Simmons et al., 2009)).

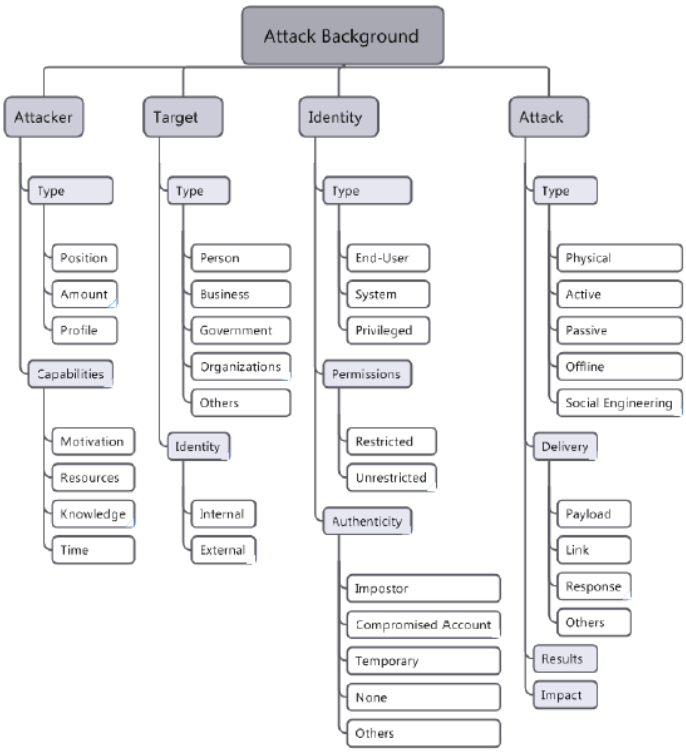


Figure 44 - TaxIdMA: Attack Background (Pöhn & Hommel, 2022)

In (Shalyapin & Zhukov, 2015) we have an example of how classification models can be used to refine cyber incident response strategies. Here incidents are compared with previous classes by creating a measure of similarity based on selected qualities. In this way, previous mitigation actions can inform the approach required to mitigate a current incident.

In (Van Heerden et al., 2016) the authors seek to visually classify cyber-attacks in South Africa. Each classification dimension shown below) is provided with three or four subcategories. The author then provides a graphical representation of twelve cyber-attacks for each of the dimensions

- Attacker
- Goal
- Mechanism
- Effect
- Motivation
- Target
- Vulnerability
- Scenario

An example of output for the 'Attacker' dimension is given below, interestingly this approach allows an attack to be described within multiple subcategories.

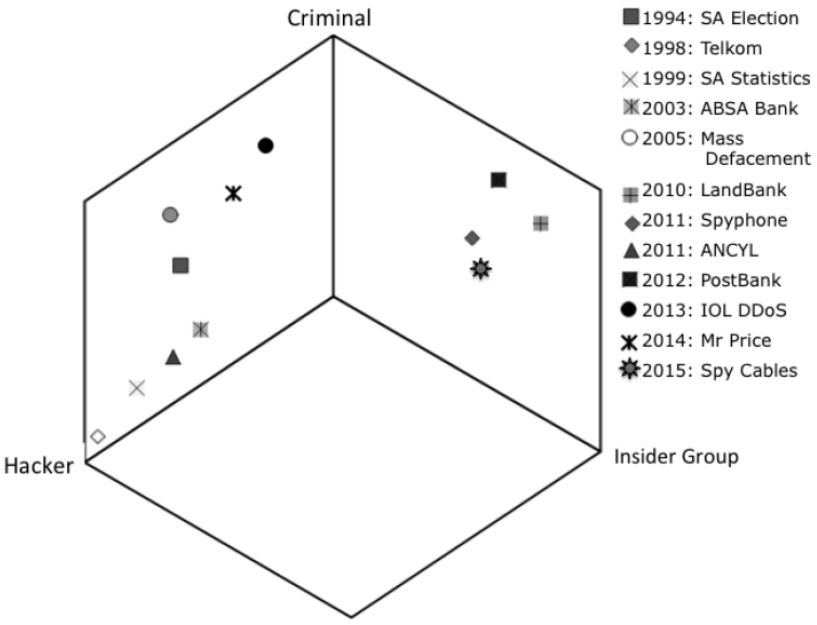


Figure 45 - Attacker Classification Graph (Van Heerden et al., 2016)

In (Derbyshire et al., 2018) we have an "analysis of cyber security attack taxonomies". Recognising that the landscape of cyber-attacks changes over time it argues that a new review of existing taxonomies is required alongside some investigation of how effective these are/remain. This paper takes a structured approach by studying the component structure of attacks and identifies elements relevant to possible inclusion within an appropriate taxonomy. After nominating several previous taxonomies an assessment framework is proposed (again based on previous proposals) and assessed against twenty example attacks. Here the authors propose the following assessment criteria alongside details of how those criteria will be assessed.

- Accepted
- Complete/exhaustive
- Comprehensible
- Mutually exclusive
- Repeatable
- Terms well defined
- Unambiguous
- Useful
- Versatile (adapts to the changing landscape)

- Human representative (can include coverage of entirely human based attacks)

The last two items are added for the 'classification of complex socio-technical systems/attacks'.

The paper finds that CAPEC is the only taxonomy able to adequately classify all twenty attack examples (even if they did not always meet all the criteria above). They also note that classification taxonomies tended to perform better the later it had been developed, this was likely due to an increased understanding of cyber-attacks over time. Despite CAPECs success in classification (potentially indicating that the ongoing development of detail and finer grain granularity was becoming comprehensive) it needed a high level of expertise to use.

In (Bahrami et al., 2019) the authors analyse forty attacks to develop a taxonomy for Advanced Persistent Threat features. This taxonomy is based around the Cyber Kill Chain and is shown below. As noted by the authors this taxonomy is currently constantly evolving as more APTs are analysed, however it is included here as a potential source of taxonomy dimensions that may be considered while developing a cyber-attack taxonomy

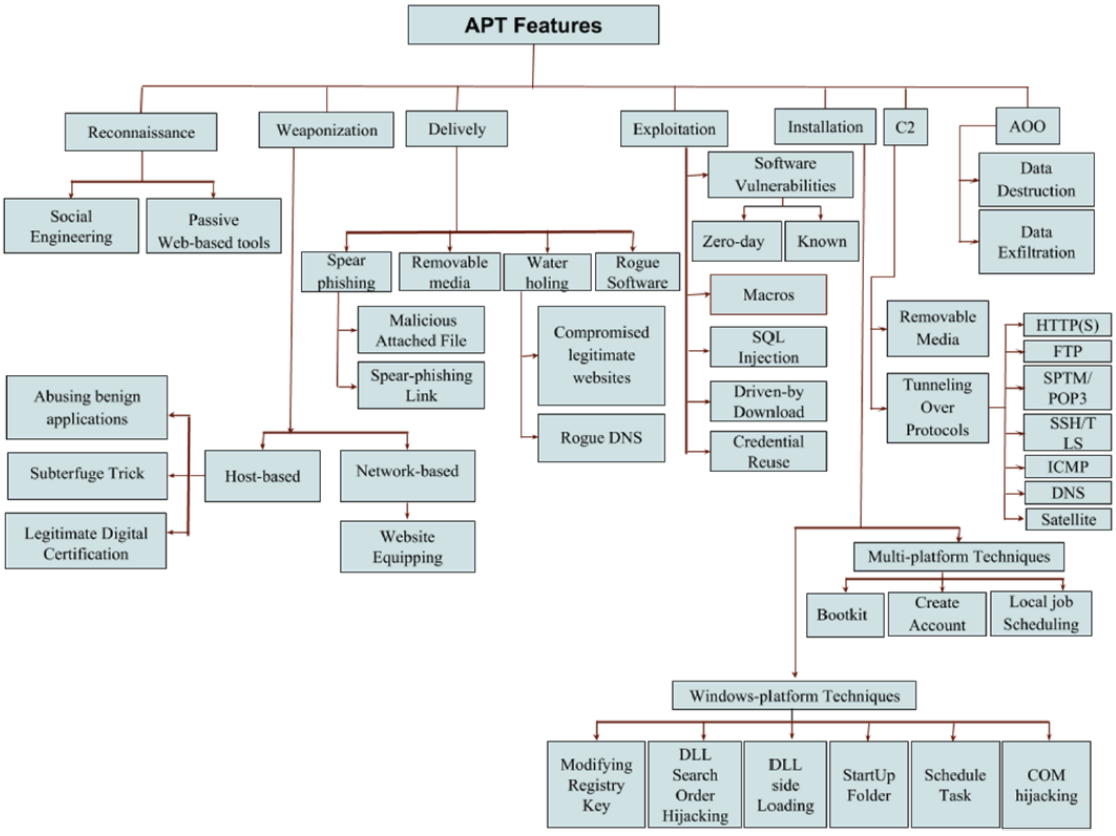


Figure 46 - CKC-based taxonomy of APT features (Bahrami et al., 2019)

3.3.3 Conclusion

The section above describes several approaches that have been proposed for cyber-attack classification. This review and conclusions will be used to guide a classification taxonomy in the following sections.

As described in (Hansman & Hunt, 2005) we will take a dimension based approach.

Although very comprehensive we will reject (Barnum, 2008) due to its complexity. We will also reject (Meyers et al., 2009) as being too simplistic and also included (at high level) in other proposals. (Uma & Padmavathi, 2013) provides more of an overview of the types of classification models so this was less useful for the purposes here. Specific classifications such as (B. Zhu et al., 2011) (S. Kim et al., 2019) are not specifically suitable for the purposes of this work, but nevertheless the key dimensions can be reviewed for relevance. (Shalyapin & Zhukov, 2015) provides an approach for comparing new attacks with previous attacks but the dimensionality is not specific. Despite the success of the CAPEC evaluation in (Derbyshire et al., 2018), this is a detailed technical classification system and too specific for use here. It may add value to record known CAPEC patterns recorded in the attack reports but this will not be pursued at this point (noted as possible Future Work).

There are a number of proposals for candidate dimensions described in (Hansman & Hunt, 2005) (Simmons et al., 2009) (Van Heerden et al., 2016) (Derbyshire et al., 2018). These will be explored further in developing a meta data model for a description of attacks using ATT&CK techniques.

A number of the referenced papers also outline requirements for a ‘good’ taxonomy that can be used for validation (in particular (Hansman & Hunt, 2005) and (Derbyshire et al., 2018)).

As noted in (Hansman & Hunt, 2005) there are many different overarching definitions of a cyber-attack. Surprisingly there seems limited attention this definition across the documents, presumably because the focus is on the taxonomy itself providing specific ‘explanation’.

In (Uma & Padmavathi, 2013) we have “exploitation of cyberspace for the purpose of accessing unauthorized or secure information, spying, disabling of networks and stealing both data and money is termed as cyber-attack”. In (Van Heerden et al., 2016) we have “any offensive manoeuvre performed against an IT system by an internal or external party is considered to be a cyber-attack”. In (Derbyshire et al., 2018) we have “we can consider a cyber-attack to be an offensive action taken against a target’s cyber infrastructure. This includes connected computers,

software, networks, procedures, and people". In (Pöhn & Hommel, 2022) we have "The use of an exploit by an adversary to take advantage of a weakness with the intent to achieve a negative impact". From (Simmons et al., 2009) we also have "A blended attack exploits one or more vulnerabilities to perform an attack against a target".

Together these can be seen to give an approximate general definition we can use within this document. So here I will use - **A cyber-attack is an offensive action taken against a target's cyber infrastructure. This includes connected computers, software, networks, procedures, and people**. This definition includes the ability to include attacks based entirely on social engineering.

3.4 Approaches to Automatic Kill Chain Detection

3.4.1 Introduction

This section presents a review of relevant literature related to the detection of APT attacks and in particular kill chains.

3.4.2 Summary Overview

The wider subject of general attack detection and prediction is extremely broad, and several surveys provide a starting background, a few specifically illustrative examples are outlined briefly here. An earlier survey is provided in (Singh & Silakari, 2009) providing a summary of detection systems - differentiating between Host Intrusion Detection Systems (HIDS), Network Intrusion Systems (NIDS), and detection analysis approaches – such as misuse detection and anomaly detection. The authors also outline a few systems current at the time of the papers publication, including signature-based systems. In (Husák et al., 2019) the authors consider developments in four areas attack projection, intention recognition, intrusion prediction, and network security situation. A review of existing and future directions (with particular interest in General Adversarial Networks (GAN)) is considered in (Soleymanzadeh & Kashef, 2022). In (Wei et al., 2021) the authors focus on APT attack detection specifically in Industrial Control Systems (ICS). The paper provides a summary of key issues and approaches however is very general in its conclusions.

Approaches to analyse configurations to more formally understand the risk of attacks (such as (Abraham & Nair, 2015)) and automated creation of attack graphs (such as (Brazhuk, 2021) linking 'the ATT&CK, CAPEC, CWE, CVE security enumerations') can be considered as part of the scope of attack detection. It is broadly the area of attack detection and projection that is of interest here and a few representative examples are given below.

An earlier proposal is given in (Cheng et al., 2011) where the Judge Evaluation of Attack intensioN (JEAN) system is developed (see also LCSS above). This is a signature-based system that uses network alerts to project probabilities of multi-stage attacks through comparison of previous and actual 'multi-stage attack session graphs (ASG)'.

In (Bhatt et al., 2014) the authors offer consideration of a framework toward detection of multi-stage APT attacks. They identify three major components, a multi-stage attack model – kill chain, a layered security architecture – a layered model increases chances of detection and, a security event collection and analysis system – alert correlation. Using this approach, the authors show how they link alerts to specific phases of an attacks within the attack model.

In (Wilkens et al., 2021) the authors investigate multi stage attacks and producing a graphical summary, which they call APT scenario graphs. Here the graph nodes represent host systems and the edges APT activity. This is with the intention of reducing data overload on the cyber analysts. The authors note that advanced alert correlation approaches (integrating alerts based on commonality such as IP address, timings etc) also seek 'to reconstruct complex attack scenarios consisting of multiple distinct steps or stages' (c.f. (Barzegar & Shajari, 2018), (Haas & Fischer, 2018)). These approaches have been found to work well for temporally/spatially well correlated data but struggle with stealthy attacks operating over long time periods. The authors develop a finite-state machine model based on the UKC (this is the Kill Chain State Machine (KCSM). Ultimately this is used to connect correlated events to a KCSM. Two examples from this paper are shown below

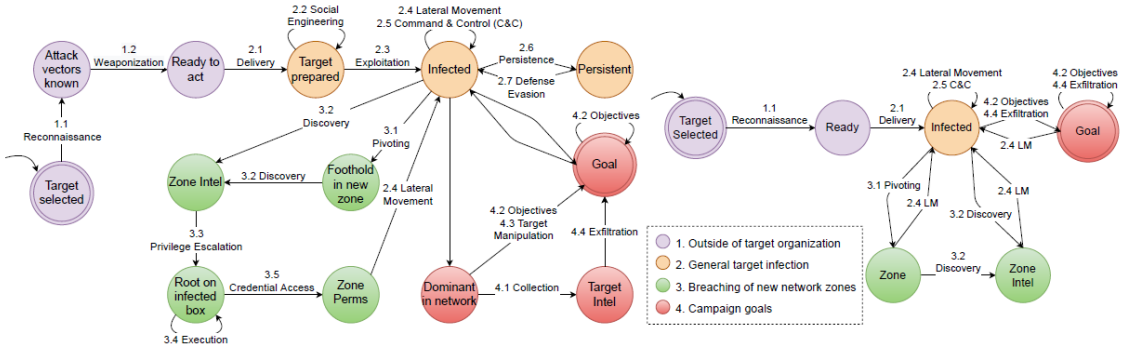


Figure 47 - KCSM Examples (Wilkens et al., 2021)

Another approach is given in (R. Zhang et al., 2017). Here, rather than correlating alerts to a generalised attack model, the authors mine IDS security logs to extract attack sequences and use these sequences to guide and support future APT detection. An example sequence is shown below

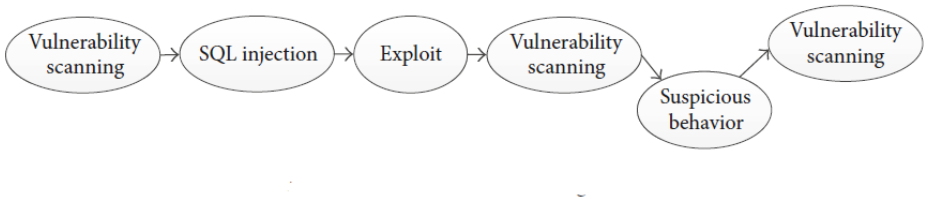


Figure 48 - Attack Sequence Mined from IDS Log (R. Zhang et al., 2017)

Above we reviewed research for justifying the assumption that a system can detect kill chains (albeit with varying levels of False Positives). Here we will focus on the ability to reason Tactics as well as Techniques (Google Scholar – "ATT&CK tactics from techniques").

In (Mireles et al., 2016) a framework is proposed to extract 'narratives' of attacks from 'traffic datasets'. The low-level information is from these datasets is ultimately used with Mandiant's Kill Chain model to build these narratives. The authors of (Lin et al., 2021) consider how to link NIDS rules to discover Tactic, Techniques and behaviours within a kill chain. Here a mixture of text mining and machine learning is applied. Results compare favourably with rcATT ((Legoy et al., 2020), see also above) and the system is intended to provide cyber analysts with additional insight as well as intelligence to assist in alert correlation.

The authors of (T. Li et al., 2020) provide some critique of existing multi-stage attack plan recognition. They note that current systems struggle with incomplete data availability because of a failure to include sequences alongside causal associations created through correlation analysis. They map attack phases into an HMM model mapping alerts to the attacker's intent and achieve probabilistic reasoning through a 'Loopy Belief Propagation' (LPG) model to reduce false positives.

In (Kurniawan et al., 2021) a prototype has been developed demonstrating the linking low-level threat alerts being to a knowledge graph and subsequent identification of Tactics. In (Kurniawan, Ekelhart, Kiesling, Winkler, et al., 2022) this is subsequently further developed toward VloGraph a virtual log Knowledge Graph from heterogeneous raw log sources across multiple hosts. Three scenarios are tested including "Scenario III—Threat Detection and ATT&CK Linking". As suggested, this shows the ability to link attack scenarios to ATT&CK techniques but also tactics. Another approach (from the same author) but for "tactical attack discovery" based on audit data is demonstrated in (Kurniawan, Ekelhart, Kiesling, Quirchmayr, et al., 2022).

The MITRE Cyber Analytics Repository (CAR) initiative should also be noted (MITRE, 2022g). This provides a knowledgebase to linking analytics to detection of ATT&CK Techniques.

Again based around an HMM in (Shawly et al., 2018) and (Shawly et al., 2021) the authors discuss an approach to dealing with detection of multiple interleaved attacks.

3.4.3 Conclusion

It is theoretically possible that future systems could detect at least fragments of an APT attack expressed as a sequence of ATT&CK Techniques. These future systems should also be capable of detecting ATT&CK Tactics.

3.5 Attack Modelling Languages

3.5.1 Introduction

A brief overview of attack modelling languages to provide background to the problem being investigated here.

3.5.2 Summary Overview

There are several complex languages that have been developed to describe low-level potential attack scenarios.

In (Eckmann et al., 2002) the authors present STATL. This is an early definition of an extensible state/transition attack description language. This describes attacks as a sequence of actions (with relevant state transitions) taken by an attacker in a non-domain specific way. This language is intended for use with intrusion detection systems to detect active attacks.

Over and above manual attack graphs and semi manual approaches such as MulVal (c.f. (Homer et al., 2013)), a notable example includes the Correlated Attack Modelling Language described in (Cheung et al., 2003). This work was driven by the desire to go beyond the linking individual alerts (primarily IDS) to individual attack steps and leaving the correlation of all such alerts to a largely manual process. In order to do this, the authors develop an abstract language building on the Intrusion Detection Message Exchange Format (IDMEF) (Debar et al., 2007). CAML allows a user to define a set of modules that define a correlation framework. Without describing the full detail, these modules are built of three main sections. Activity – Specifying the events that will trigger this module. Pre-Condition – Is used to place additional constraints that may limit triggering this module. Post – Is used to define inferences to made in this situation. In this way a framework can be built to correlate events toward recognised attack steps, techniques and structures using the Attack Pattern components within CAML. The paper then describes how to ‘implement a scenario

recognition engine'. Here the authors integrate CAML specifications with an expert system Production-Based Expert System Toolset (P-BEST) (Lindqvist & Porras, 1999) to recognise and forward plan attacks.

In (Johnson, Pontus; Lagerstrom, Robert; Ekstedt, 2014) the authors present MAL (Meta Attack Language). This language allows an expert to use domain-specific knowledge to generate domain-specific attack modelling languages for more general users. These may in turn be used to semi-automatically create attack graphs that can be used to assess the cyber security of systems MAL allows security experts to codify domain-specific knowledge. The language presented allows users to define classes containing attack steps. Attack steps may have types of AND and OR allowing the construction of multiple possible attack paths through the graphs. The model also supports timings via probability distributions representing expected time taken by an attacker. The language also supports inheritance for reuse of components. An example implementation using MAL is shown in (Xiong et al., 2022). Here the authors present a threat modelling language (enterpriseLang) based on the MITRE Enterprise ATT&CK Enterprise Matrix. The ATT&CK techniques and assets are linked and converted into MAL language files ultimately allowing the construction of trees/graphs of potential attacks. The capability of enterpriseLang is exercised against two example attacks to demonstrate that the relevant attack sequences can indeed be found in the generated model.

3.5.3 Conclusion

The languages discussed here are primarily used to model a set of potential attacks on a system. This is to be used by expert analysts to investigate overall cyber security of a configuration. A much simpler (at least in terms of scope) area of investigation exists in how to record the actual sequence of steps taken by an attacker in a specific attack and making these available in a more machine-readable form than the common openly available textual intelligence reports.

3.6 Sequence Comparison

3.6.1 Introduction

Readable overviews of relevant sequence matching techniques is provided in (Studer & Ritschard, 2016) and also (Hosangadi, 2012). This includes a practical quote "Even if some distance measures underperform, the study shows that there is no universally optimal distance index, and that the choice of a measure depends on which aspect we want to focus on".

This paper identifies important elements of a sequence, these include:

- Experienced states – In this case the techniques (that form 'the alphabet')
- Sequencing – The order of the states

The following elements relate to time which tends not to be included in the open-source reports used here. At this point I am ignoring these but will note the time dimension in future work proposals.

- Distribution – The total time spent in each state (within the sequence)
- Timing – The time at which each state occurs
- Duration – The consecutive time spent at each state.

This paper also identified three main categories of dissimilarity measures

- Distances between distributions
    - We cannot investigate these with the data available as we do not have timings
- Counting common attributes in sequences
- 'Cost' to transform from one sequence to another (or Optimal Matching).

So, a few different measures were explored.

For 'Counting common attributes in sequences', possible major approaches are:

- Simple Hamming Distance
- Length of the Longest Common Subsequence
- Number of Matching Sub-sequences

For ''Cost' to transform from one sequence to another' (or Optimal Matching), possible approaches are:

- Needleman and Wunsch (1970)
- Generalised Hamming
- Levenshtein II

After some very simple experimentation into Levenshtein, this was abandoned. Transformation cost approaches are abandoned as they do not seem sensitive to the structure of the cyber-attacks. The results obtained did not provide a clear similarity measure.

Based on common techniques used in matching genetic strands (e.g. outlined in (Chan, 2007) and (MIT, 2011) the standard FASTA and BLAST algorithms were also investigated. These may benefit from further examination but on initial inspection seem to be transformation cost-based approaches.

Another possible line of investigation also includes the mining of attack behaviour patterns (e.g. (A. F. Zhang et al., 2007)). Using the well-known Apriori data mining technique (Agrawal & Srikant, 1994) we can analyse the observed technique streams and identify association rules. This can then be used to match sub-sequences and make possible decisions about next steps.

Additionally, a lot of detailed research papers seem mostly focussed on developing efficiency for known algorithms when applied to the very large datasets used in DNA sequencing. This is not an issue in this application example (at this point), so accordingly I have limited investigation to the base approaches and Longest Common Subsequence (LCSS).

3.6.2 Longest Common Subsequence (LCSS)

An initial understanding of LCSS can be see through a simple example (relevant to this discussion).

Consider a sequence of techniques representing an attack. Shown below

T1566.001 / T1204.002 / T1203 / T1102.002 / T1071.001 / T1105 / T1059.003 / T1083 / T1082
T1016 / T1007 / T1069.001 / T1049 / T1105 / T1059.003 / T1036.005

Also consider a sequence of techniques observed within a system. Shown below

T1566.001 / T1204.002 / T1059.003 / T1083 / T1082 / T1016 / T1007 / T1069.001 / T1049

Comparing the two sequences and moving from left to right we have two contiguous sub sequences (that also maintain shared order between the two sequences)

T1566.001 / T1204.002

and

T1059.003 / T1083 / T1082 / T1016 / T1007 / T1069.001 / T1049

We then say we have a total of nine common elements (in the correct order). That is LCSS has length 9

This a more direct approach than generalised Hamming approaches (e.g. (Bookstein et al., 2002) and (Moreau et al., 2022) however development of fuzzy matching in this area may provide a direction for future work).

Here is the general description of LCSS from (Gusfield, 1997) and as described in (Wikipedia, 2023b)

Let two sequences (using alphabet S) be defined as follows:

$X=(x_1 , x_2 , \dots x_n)$ and $Y=(y_1 , y_2 , \dots y_n)$

The prefixes of X are $X_0, X_1, \dots X_n$

The prefixes of Y are $Y_0, Y_1, \dots Y_n$

Let $LCS(X_i,Y_j)$ represent the set of common subsequence of X_i and Y_j

This set of sequences is given by

$$LCS(X_i, Y_j) = \begin{cases} \epsilon & \text{if } i = 0 \text{ or } j = 0 \\ LCS(X_{i-1}, Y_{j-1})\hat{}\, x_i & \text{if } i, j > 0 \text{ and } x_i = y_j \\ \max\{LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)\} & \text{if } i, j > 0 \text{ and } x_i \neq y_j. \end{cases}$$

Relevant LCSS applications are discussed a little further in (C. Chen & Qin, 2009) and (Du et al., 2009). A critique and extension to LCSS (JEAN) is developed in (Cheng et al., 2011). JEAN provides increased fuzzy matching capability. but for simplicity (as this is not the main purpose of this work) the basic LCSS algorithm is used for demonstration here.

A dynamic programming approach can be implemented in Python, and this has a complexity order of

$$O\left(N\prod_{i=1}^{N} n_i\right).$$

3.6.3 LCSS as a Distance/Similarity Measure

There are a number of proposals around using LCSS as a dissimilarity (or distance) measure.

Although investigating similarity in multidimensional trajectories (Vlachos et al., 2002) investigates the use of LCSS as a non-metric similarity measure (this is motivated by a desire to be more resilient to noise).

$$dist_{LCSS}(A,B) = 1 - \frac{|LCSS(A,B)|}{n}$$

Here n the size of the smaller sequence min(|A|,|B|)) is proposed.

In a study of several time series classification algorithms (Bagnall et al., 2017) it is noted that ensemble measures provide the basis of a better classification models (see also below). This paper also confirms the effectiveness of Dynamic Time Warping (DTW), however in the sequences we are investigating we are currently missing the timing elements, so the benefits provided by this approach are not relevant here.

In (Coco & Keller, 2012) another distance measure based on LCSS is referenced "Once the longest subsequence is found, the similarity score is calculated as the ratio between the length of longest common subsequence and the geometric mean of the lengths of the two sequences. The resulting values range from 1 for most similar to 0 for least similar"

$$\textbf{LCS Similarity} \quad \frac{\mathbf{L_{ength}(LCS)}}{\sqrt{\mathbf{L_{ength}(SP\ 1) * L_{ength}(SP\ 2)}}}$$

This is the measure that I will use in this work.

The reason for noting the validity of these measures is to illustrate that we now have a measure of similarity that can be used to both rank the matching and be used for clustering and classification

In Future Work I will also briefly describe another research area that can be motivated by this clustering and classification, and this will be around the possibility of generating generalised attack patterns that can be used to provide a more generic set of proposals for responses to cyber analysts.

3.7 Markov Models

3.7.1 Introduction

A summary based on (Visser & Speekenbrink, 2022) to provide context later in this document.

3.7.2 Markov Chains

Markov Chains (or Models) are used to calculate probabilities in chains of events (or state changes).

These chains assume that in the process we are modelling the next state depends only on the current state.

If we have a set of possible states q_1, q_2, … q_i

This assumption can be represented as follows

$P(q_i=a|q_1 \dots q_{i-1}) = P(q_i=a|q_{i-1})$

The key components of a Markov Chain are:

A set of **N possible states** within the system

$Q=q_1, q_2, \dots, q_N$

A **transition probability matrix A**

$A=a_{11}, a_{12} \dots a_{1n} \dots a_{nn}$

Here each a_{ij} represents the probability of transitioning from state *i* to state *j*

And the sum of the matrix row $\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$

An **initial probability distribution π**

$\pi = \pi_1 , \pi_2 , ... , \pi_N$

Here π represents the probability that the Markov chain will start in state i

Also, we have $\sum_{i=1}^{N} \pi_i = 1$

3.7.3 Hidden Markov Models

Assume that we have a system that can be modelled as a Markov Process.

Assume also that there are a set of hidden states that cannot be observed directly, but that we can infer knowledge about these hidden states via observable states.

More directly we have a two-level process, an example is illustrated below

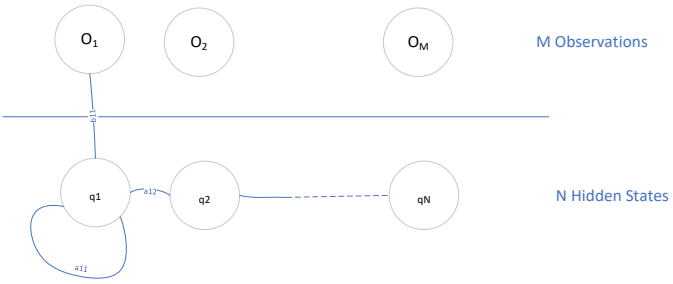


Figure 49 - Hidden Markov Model Process

In addition to the elements described above we also have

A **set of M distinct observation** 'symbols'

$V = \{ v_1, v_2, ... v_M \}$

An **observation** sequence (each observation is a symbol from V)

$O = o_1, o_2, ... o_T$

An **observation (or emission) probability matrix**

$B=b_{11}, b_{12} ... b_{1n} ... b_{nm}$

Here each b_{ij} represents the probability $q_i(o_j)$ that is the probability of observing symbol o_j when in state q_i ($i \in [1,N]$, $j \in [1,M]$)

And in particular $\sum_{j=1}^{n} b_{ij} = 1 \; \forall i$

The HMM is commonly represented as a 3-tuple (A, B, π), where A is the state transition matrix, B is the observation probability matrix, and π is the initial probability vector.

As described in (Dass et al., 2021) (III) three key problems can be addressed using Hidden Markov Models

- The Evaluation Problem
  - Given a set of observations O and HMM (A, B, π), how likely is it that the HMM results from these observations.
- The Decoding Problem
  - Given a set of observations O and HMM (A, B, π), find the hidden state sequence that best fits these observations
- The Optimization Problem
  - Given HMM (A, B, π) how can we optimize A, B, π to maximize P(O| HMM).

The Decoding Problem may be tackled using the **Viterbi Algorithm**. The Viterbi Algorithm is a dynamic programming approach (brute force is too inefficient) to find the most likely state sequence (here HMM = λ) (Yannakoudakis, 2018)

$$\hat{X} = \underset{X}{\operatorname{argmax}}\, P(X, O|\, \lambda)$$

$$= \underset{X}{\operatorname{argmax}}\, P(O|\, X, \lambda)\, P(X|\, \lambda)$$

$$= \underset{X_1 \ldots X_T}{\operatorname{argmax}} \prod_{t=1}^{T} P(O_t|\, X_t)\, P(X_t|\, X_{t-1})$$

3.8 The Challenge

As stated in (Ahmed et al., 2021) much research into APT activities is constrained by a lack of openly available data. This is also discussed in (Lemay et al., 2018) and (Alshamrani et al., 2019) where the reports available to study details of APT attacks are limited to just a small number of well-known reports. It is also noted that much of this existing information is made available through industry as opposed to academia.

MITRE ATT&CK provides a standardisation of TTP terminology that is being embraced by cyber analysts when creating openly available reports on cyber-attacks. ATT&CK also includes descriptions of APTs in terms of the TTPs that they use. This can a useful contribution to intelligence available to a user wishing to understand cyber situational awareness and to related research themes. However, as noted in (Spring & Al-shaer, 2020) amongst others, this data does not include temporal intelligence on the related attack sequences.

Based on the literature reviews carried out across the background and related areas documented above, there has been no specific attempt to define a straightforward model to create a machine-readable record of attacks described as a sequence of ATT&CK Tactics and Techniques.

This then represents the purpose of the subsequent research here.

Chapter 4 Research Questions

4.1 Introduction

This section provides a brief overview and restatement of the broad proposal (see Proposal Overview) and also the Research Questions (see Research Questions) that have been used to further guide this work.

4.2 Proposal Overview

MITRE ATT&CK is a framework (common taxonomy) for describing the Tactics, Techniques, Tools, and Procedures used in cyber-attacks. This openly accessible knowledge base continues to develop and is based on analysis of numerous cyber-attack reports (and additional intelligence). It provides structured descriptions of several Tactics and Techniques.

APTs and Tools used in attacks are then described in terms of these Tactics and Techniques and examples of APT procedures related to the Techniques.

As well as providing a common language to describe tactics and techniques this knowledge based also provides intelligence that can be used (e.g., by Red Teams) to model the behaviours of various attackers and test the cyber defences for a site.

This suggests the potential to use the records of previous attacks by APTs, described in terms of ATT&CK, to help cyber defences detect patterns of TTPs that may indicate an APT attack.

In practice the descriptions of APTs provided in ATT&CK have been ‘simplified’. Individual attacks are not documented in detail. Additionally, the APT descriptions are provided as a list of techniques that have been observed as being used by that APT. There is no indication of the sequencing of the TTPs used in various attacks. This sequencing can provide additional assistance to analysts when considering observed techniques if they also have knowledge of the order in which they were detected.

This work provides a contribution to address the lack of available attack sequence intelligence described as a sequence of MITRE ATT&CK TTPs. An example approach showing how software may read and analyse this intelligence is included.

It does this in the following way.

- Firstly, an approach is developed to model cyber-attacks as sequences of TTPs. To our knowledge a similar model has not been defined in any research literature.
    - This model includes:
        - Both a categorisation (of the attack types) and sequencing model (with discussion on a justification and limitations for this sequential attack model view).
        - Provision of a link between the MITRE ATT&CK Tactic and Unified Kill Chain (UKC) model descriptions (to support further study of Kill Chain based attack analysis).
        - An approach to chaining sequences together to describe multi-step attacks.
- Secondly, this model is exercised using a representative set of example attack sequences drawn from both open-source attack reports referenced through MITRE and additionally researched open-source reports.
    - The example set used is justified through comparison with similar types of research papers and compares favourably in terms of coverage.
    - Currently 26 attacks with 390 event steps across 97 different Techniques are codified. The attack examples cover all relevant ATT&CK Tactics and UKC phases. It includes both single and multi-step attacks.
- Thirdly, the modelled example attack sequences are used to demonstrate applications of how the data can be used by software to investigate the sequences observed
    - This includes (see also code and data in (Maidens, 2023):
        - An extensive python code base used to load ATT&CK data and store in a relational model (see (Maidens, 2023))
        - The data is also persisted in a lightweight python NetworkX graph database (NetworkX is a Python package for graph database manipulation) and stored in an open XML form (gexf).

- Fourthly, several examples of usage (implemented in python) are presented
    - Demonstration that sequencing of attack TTPs improves the ability to distinguish between the behaviours of different APTs (see also Simple Demonstration of Additional Intelligence Provided).
    - Demonstration of a pattern matching approach to compare observed sequences with existing attack signatures (see also Using the Attack Model – LCSS Fragment Matching)
        - This approach has used LCSS. A straightforward approach but used extensively in DNA sequencing and resilient to missing observations in detection systems
        - Novelty in this proposal includes multiple levels of LCSS. Tactic sequence LCSS followed by Technique Sequence LCSS (this is to enhance distance measures by treating Technique sequences with related Tactic level sequences as similar)
    - Demonstration of how data in this form can be used as input to a Hidden Markov Model (see also Using the Attack Model – Hidden Markov Model)
        - Attacker's behaviour is understood as stochastic and a sequential attack forming a Markov chain
        - This is used to demonstrate how 'hidden' attacker Tactics can be derived from observed Technique sequences. Aiding understanding of a likely attack intentions
    - Demonstration of next step prediction issues (see also Using the Attack Model – Markov Model).
    - Discussion on linking the attack sequences to the Unified Kill Chain (UKC) (see also Using the Attack Model – Unified Kill Chain).

4.3 Research Questions

This section repeats the points outlined in Research Questions. This is just for readability and flow of this chapter.

	Main Research Questions
R1	Can the ATT&CK APT descriptions be used to support the detection of multi-step cyber-attacks and potentially anticipate next steps?
R1a	Can we record known APT attacks as sequences of ATT&CK Tactics and Techniques?

R1b	Will the sequences in R1a provide us with additional intelligence over and above the unordered lists of APT Techniques currently provided within the ATT&CK knowledgebase?
R1c	Can we provide a classification system for the sequences in R1a that will also support some further analysis of recorded attacks?
	Supporting Research Questions (Literature Reviews)
R1_SuppA	Can we create multi-step ATT&CK technique chains from sensor networks?
R1_SuppB	Can we extract interleaved attack chains from ATT&CK techniques detected from sensors?

Chapter 5 Characterising the Base Data

5.1 Introduction

This section provides a general overview of the MITRE ATT&CK Knowledge Base.

This has been included only to provide the reader with a background summary of the current ATT&CK knowledge base (used in this work). This includes an outline view of the data included and some illustration of the capabilities.

The overview includes:

- A general description of ATT&CK (see ATT&CK)
- A general description of the key subsections (data content matrices) within ATT&CK and a clarification that this work will focus on the Enterprise matrix (see ATT&CK Data Content Matrices).
- An overview of some related developments within MITRE that complement the use of ATT&CK (see Related Developments).
- A little more detail on the data model within the Enterprise matrix (see Enterprise ATT&CK Data Model)
    - Additional views of this model (and also the underlying STIX model) are also illustrated for completeness. With the exception of the Relational Model (which is used to create a locally accessible copy of the master data) these models are not used elsewhere in the document and are presented for background information only.
- A high-level summary of the actual content of the Enterprise matrix (see ATT&CK (Enterprise) Data Content High Level Summary).
- A brief investigation into clustering of APT observed use of Tactics (An Initial Attempt at Clustering Group 'Fingerprints'.
- Some concluding notes (see Conclusion)

5.2 ATT&CK

A brief overview of the purpose of the ATT&CK knowledge base, the data sources and the specific version used for this work is provided here.

"ATT&CK is a catalogue of techniques and tactics that describe post-compromise adversary behaviour on typical enterprise IT environments. The core use cases involve using the catalogue to analyse, triage, compare, describe, relate, and share post-compromise adversary behaviour." (MITRE, 2022c)

For this work the instance of the ATT&CK catalogue provided by MITRE and stored in GitHub at (MITRE, 2022c) has been used. This has been accessed through the TAXII 2.0 service (Wunder et al., 2017) (Burns & MITRE, 2018) provided by MITRE at (MITRE, 2022f). This is based on data downloaded on 25/04/2022 (version 11). Development of the MITRE ATT&CK data model is very active and continuously developing, however this release provides a relevant checkpoint for the focus on TTPs here.

The ATT&CK Cyber Threat Intelligence is curated from several sources, this includes various cyber-attack reports, internal government intelligence and other feeds such as commercial companies and information sharing groups.

This openly available Knowledge Base provides the basis for a common dictionary and taxonomy to describe the (primarily) technical behaviours being used in cyber-attacks. Additionally, it provides descriptions of a number of APTs and Tools in terms of these behaviours.

5.3 ATT&CK Data Content Matrices

For this work I am focussing only on the Enterprise matrix within the whole ATT&CK knowledge base. An overview of all the matrices within ATT&CK is given here.

There are currently three major distinct knowledgebases / matrices of intelligence content within ATT&CK (MITRE, 2019c). As described in the Threat Modelling section above (Threat Modelling Approaches 3rd paragraph) these have been refined to address differing application domains.

These are:

- Enterprise
    - This Matrix contains information for the following platforms: Windows, macOS, Linux, PRE, Azure AD, Office 365, Google Workspace, SaaS, IaaS, Network, Containers.
- Mobile
    - Tactics and techniques representing the two MITRE ATT&CK® Matrices for Mobile. The Matrices cover techniques involving device access and network-

based effects that can be used by adversaries without device access. The Matrices contains information for the following platforms: Android, iOS.

- Industrial Control Systems (ICS)
    - ATT&CK for ICS is a knowledge base useful for describing the actions an adversary may take while operating within an ICS network. The knowledge base can be used to better characterize and describe post-compromise adversary behaviour. Please see the overview page for more information about ATT&CK for ICS.

5.4 Related Developments

ATT&CK has been developed in parallel with a number of related undertakings. Some of the key ongoing MITRE developments are summarised here.

5.4.1 Cyber Analytics Repository (CAR)

The Cyber Analytics Repository (CAR) is a parallel development alongside ATT&CK (MITRE, 2022g).

This provides a knowledge base of analytics (as pseudocode) that can be implemented to provide alerts about ATT&CK Technique within the monitored environment.

Implementation of these analytics is developing and being implemented in various environments. They are supplied (with varying coverage) as Sigma, Splunk and Elastic Detection rules.

5.4.2 MITRE D3FEND

A parallel development D3FEND (MITRE, 2022h) (Kaloroumakis & Smith, 2021) seeks to build a knowledge base linking cyber threats to countermeasure components and capabilities. “The graph contains semantically rigorous types and relations that define both the key concepts in the cybersecurity countermeasure domain and the relations necessary to link those concepts to each other”.

This knowledge base of countermeasures is linked to ATT&CK through ‘Related ATT&CK Techniques”. This is illustrated with a simple example (from (MITRE, 2022h)) below:

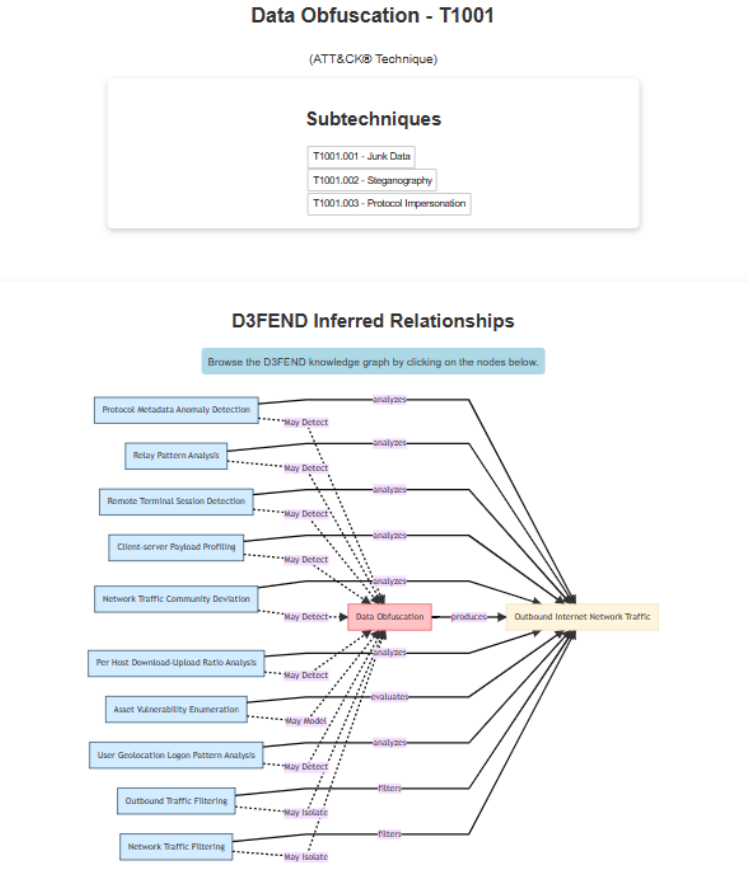


Figure 50 - D3FEND ATT&CK Relationship Example

5.4.3 Other Examples

The Attack Hypothesis Generation tool proposal discussed in (Elitzur et al., 2019) recognises that previous attacks can provide defenders with useful intelligence about possible future attacks. To contribute to this space, they propose a tool to use existing intelligence (in this case ATT&CK) to generate a knowledge graph to can be used to hypothesise possible attacks that may occur on the target organisation.

5.5 Enterprise ATT&CK Data Model

This section provides an overview of the current (at time of writing) ATT&CK data model. Various additional views of this model are also summarised in this section to provide further background (see also summary of objectives in this chapter above in Introduction).

In High Level Overview the explicit named relationships implemented within this model are shown.

In As a Subset of the STIX2 it is shown how ATT&CK is related to the STIX2 (see also Approaches & Standards). An important standard used for the sharing of threat intelligence.

Also in As a Subset of the STIX2 model examples such as Unified Cybersecurity Ontology (UCO) (Syed et al., 2016) are referenced. These are intended as illustrations of attempts at processing multiple cyber threat information sources through a common knowledge graph.

The relational model presented in A Relational View was developed explicitly for this work. It is used to underpin python code (Maidens, 2023) that was developed to execute the analysis included in this work. Due to the well understood structure of a relational model it is hoped that future availability in this form could help others to perform additional future analysis (see also Future Work).

The graph form (see section A Graph View) briefly presented here was developed in this work but was only used temporarily and remains incomplete. It was initial work for consideration of a formal ontological model that could be connected to additional threat information sets (e.g. CAPEC, CWE and NVD). This could well be developed further in the future but became beyond the scope of this work. It was also temporarily used as input to provide the ATT&CK data in Prolog form. This was then used as input to a parallel Argumentation Based Reasoner development based on a previous paper (Karafili et al., 2018).

Also in A Graph View, two example attempts at describing STIX as an ontology ((E. Al-Shaer & Chu, 2017) and (Ulicny et al., 2014)) are shown. Both of these ontologies were developed as part of work looking into approaches that can improve how we may move more efficiently from 'threat intelligence' to 'mitigation actions' (E. Al-Shaer & Chu, 2017) (including better visualisation for human analysts). Broadly, they were developed to provide foundations for reasoning across multiple sources of cyber threat intelligence within wider frameworks.

5.5.1 High Level Overview

As described above, a high-level overview of the current ATT&CK data model and the explicit named relationships implemented within this model is shown in Figure 51 and Table 3 below

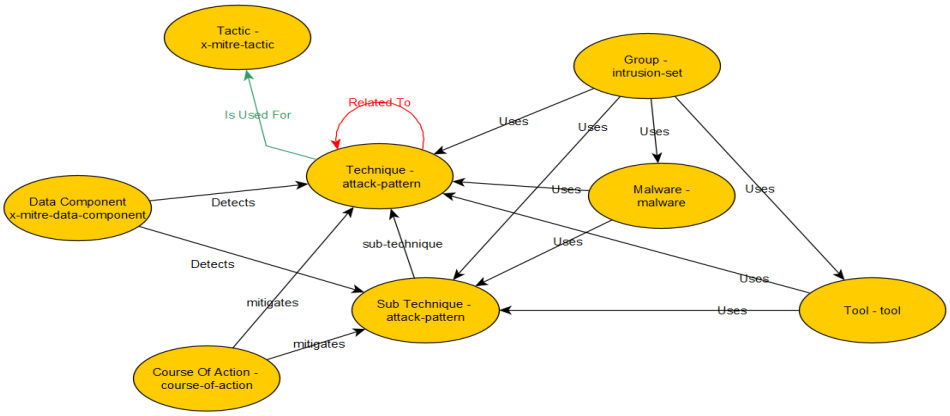


Figure 51 – ATT&CK High Level Data Model

Relationships marked red are not yet implemented in the relational model shown below (see also A Relational View). Relationships marked green are implicit (with added relationship name) within the data elements.

A tabular 'meta model' description of above is shown below.

REL_SOURCE_TYPE	REL_SOURCE_SUBTYPE	REL_TARGET_TYPE	REL_TARGET_SUBTYPE	REL_TYPE
x-mitre-data-component	x-mitre-data-component	attack-pattern	sub-technique	**detects**
malware	malware	attack-pattern	technique	**uses**
malware	malware	attack-pattern	sub-technique	**uses**
x-mitre-data-component	x-mitre-data-component	attack-pattern	technique	**detects**
course-of-action	course-of-action	attack-pattern	technique	**mitigates**
course-of-action	course-of-action	attack-pattern	sub-technique	**mitigates**
tool	tool	attack-pattern	sub-technique	**uses**
tool	tool	attack-pattern	technique	**uses**
intrusion-set	intrusion-set	attack-pattern	sub-technique	**uses**
intrusion-set	intrusion-set	attack-pattern	technique	**uses**
intrusion-set	intrusion-set	malware	malware	**uses**
intrusion-set	intrusion-set	tool	tool	**uses**
attack-pattern	sub-technique	attack-pattern	technique	**subtechnique-of**
attack-pattern	technique	attack-pattern	technique	**related-to**

Table 3 - ATT&CK Meta Model

Data Type	Description
Group	The ATT&CK dataset contains over 100 descriptions of known APTs. These are created through analysis of known cyber threat attack reports. The groups are described in terms of the techniques (including sub-techniques) and software (tools and malware) that they use.
Tactic	"Tactics represent the "why" of an ATT&CK technique. It is the adversary's tactical objective: the reason for performing an action. Tactics serve as useful contextual categories for individual techniques and cover standard notations for things adversaries do during an operation" (Bodeau et al., 2018b)
Technique Sub-technique	"Techniques represents "how" an adversary achieves a tactical objective by performing an action." (Bodeau et al., 2018b) ATT&CK supports two level of techniques. Initial reviews criticised the lack of granularity in the initial versions of the Knowledge Base. Subsequent versions included more granular sun-techniques. Following some discussion with the MITRE team supporting and developing this initiative it seems that a pragmatic decision was made to limit the schema to these two levels (to avoid the complexity that has been observed in CAPEC as it has developed) to encourage a wider take-up. This is not without its own issues, for instance in a commonly observed Technique T1059 – Command and Scripting Interpreter, a flag has been added to note that this can also be invoked remotely but intelligence collected does not actually note whether or note this was true within an attack.
Malware Tool	Software used by the attacker (through techniques and sub-techniques)
Course of Action	Advice on mitigation against techniques and sub-techniques
Data Components	Provide links to detection advice (see also MITRE D3FEND (MITRE, 2022h) and MITRE Data Sources (MITRE, 2022b))

Table 4 - ATT&CK Data Types

5.5.2 As a Subset of the STIX2

The ATT&CK intelligence can be accessed (via git repository or TAXII service) as STIX2 JSON objects. This is described in more detail in (MITRE, 2022a) which describes how the STIX2 objects are used and populated. This section provides a brief overview of how the JSON objects within ATT&CK integrate with the wider STIX standard.

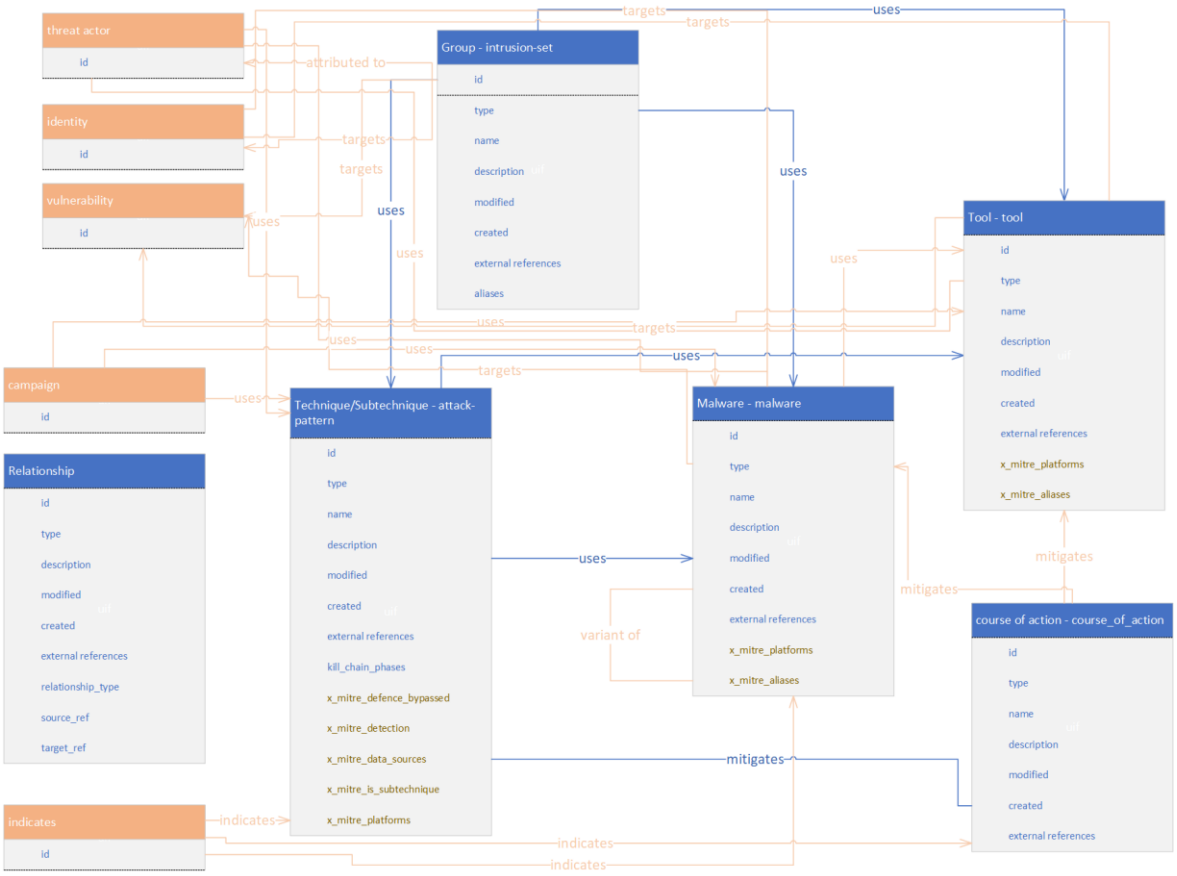


Figure 52 - MITRE ATT&CK as STIX2 model

There are a few key points to note here:

- Several relationships to possible STIX2 Domain objects included in the STIX2 specification are not implemented in ATT&CK.
    - A number of these can be extracted from the data content to improve intelligence and support for machine processing further.
- Objects in ATT&CK are augmented with custom ‘x_mitre’ elements.
    - Some of these have been added relatively recently and appear to be strengthening support for cyber situational awareness capabilities.
- The intrusion-set is used to describe the threat-actor. Relationships are then maintained from the intrusion-set to the relevant attack-patterns, tools malwares. Individual attack and campaign descriptions are not maintained.

- Limited kill chain intelligence is maintained through in-line reference to Tactics (in the kill_chain_phases element of the attack-pattern) but these are not specific for each Group.
- Work continues to develop and connect the STIX standard to more formal ontologies. For instance
    - The Unified Cybersecurity Ontology (UCO) (Syed et al., 2016) and related development at (Syed et al., 2018) (and used in (Narayanan et al., 2018) by gathering information from various sources and processing through a common processable knowledge graph).
    - Another example is given in MalOnt. This is an ontology for gathering Malware Threat Intelligence from threat reports. Discussed in (Rastogi et al., 2020) and related development at (Dutta, 2020)

5.5.3 A Relational View

There are a number of relational models of ATT&CK that have been developed (e.g. (Shallabi, 2019)). Because of the focus here, a specific model was developed to use.

Python (see (Maidens, 2023)) was used to automatically extract data from the MITRE TAXII service to implement and populate a relational model. This is then stored and accessed locally. A great deal of effort has been expended on this but has proved very useful and much more flexible than using existing (and often undocumented and unmaintained) options.

It was written with a view to being easily reusable for others on commodity laptops, but really requires some finishing work to deal with installation, maintenance, and runtime error checking (see Future Work). Other examples such as (Rodriguez, 2022) exist but are often difficult to set up or alternatively needs constant access to the TAXII server to operate.

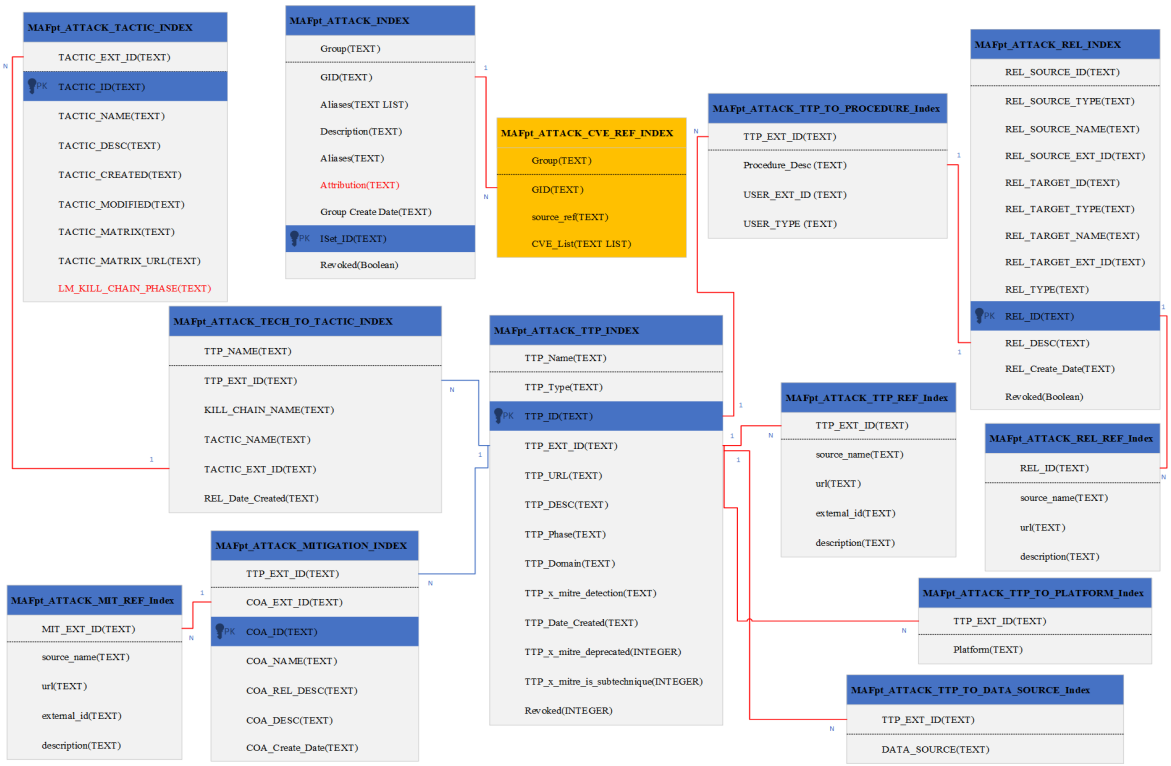


Figure 53 - Interim ATT&CK Relational Model

A number of utility functions, listed in Appendix B, have also been written to access this data (Maidens, 2023).

5.5.4 A Graph View

A graph view that was developed during early investigation stages of this work as the basis of an ontology (not subsequently developed further) is shown here

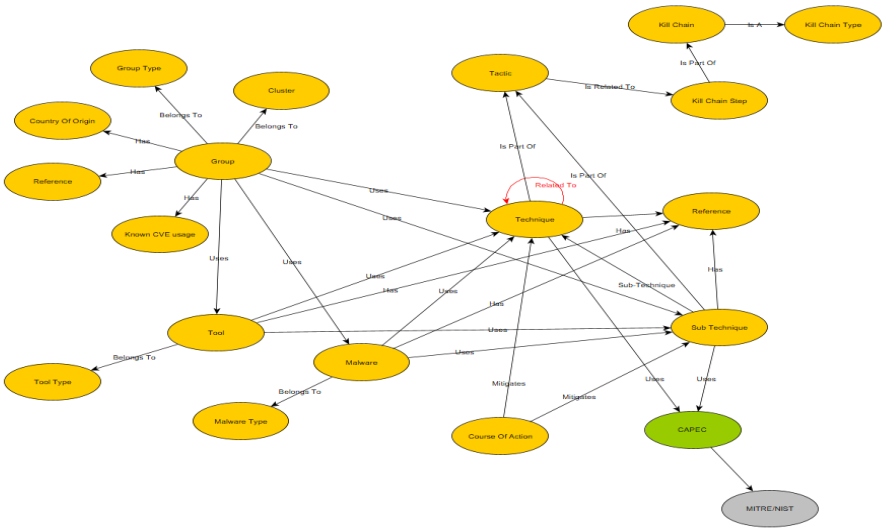


Figure 54 - ATT&CK as Graph

This model was also temporarily used to present the data as Prolog rules. This data was then used as input to a parallel Argumentation Based Reasoner development based on a previous paper (Karafili et al., 2018), however further development was not pursued after closure of the project. The illustration is provided here for background information.

Ontologies for STIX (a standard used within ATT&CK) have been expressed in both (E. Al-Shaer & Chu, 2017) and (Ulicny et al., 2014). These are visually summarised below.

Both of these ontologies were developed by the authors as part of work looking into approaches that may improve how we can move more efficiently from 'threat intelligence' to 'mitigation actions' (E. Al-Shaer & Chu, 2017) (including better visualisation for human analysts). Broadly, they provide foundations for reasoning within wider frameworks.

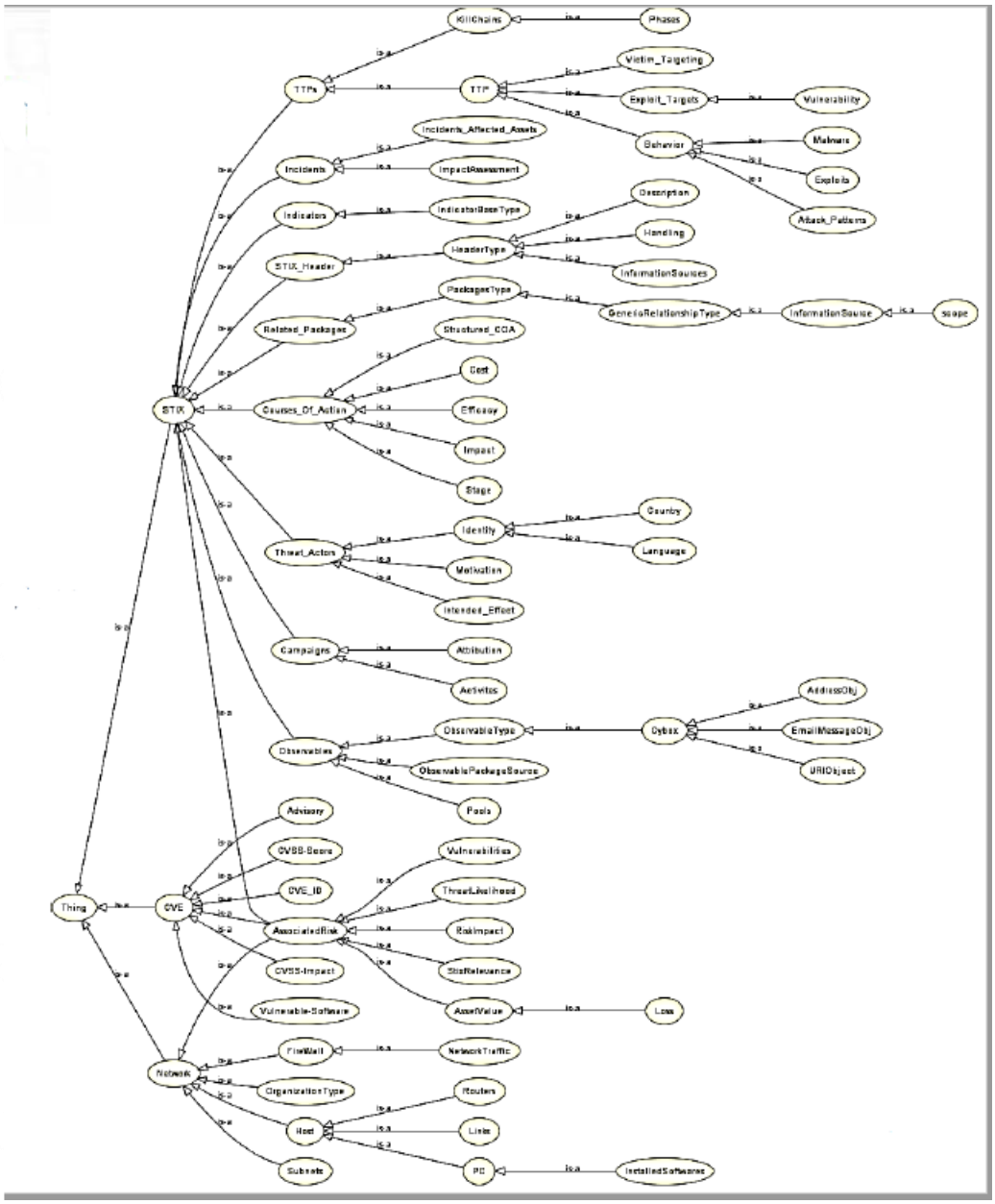


Figure 55 - STIX Ontology (E. Al-Shaer & Chu, 2017)

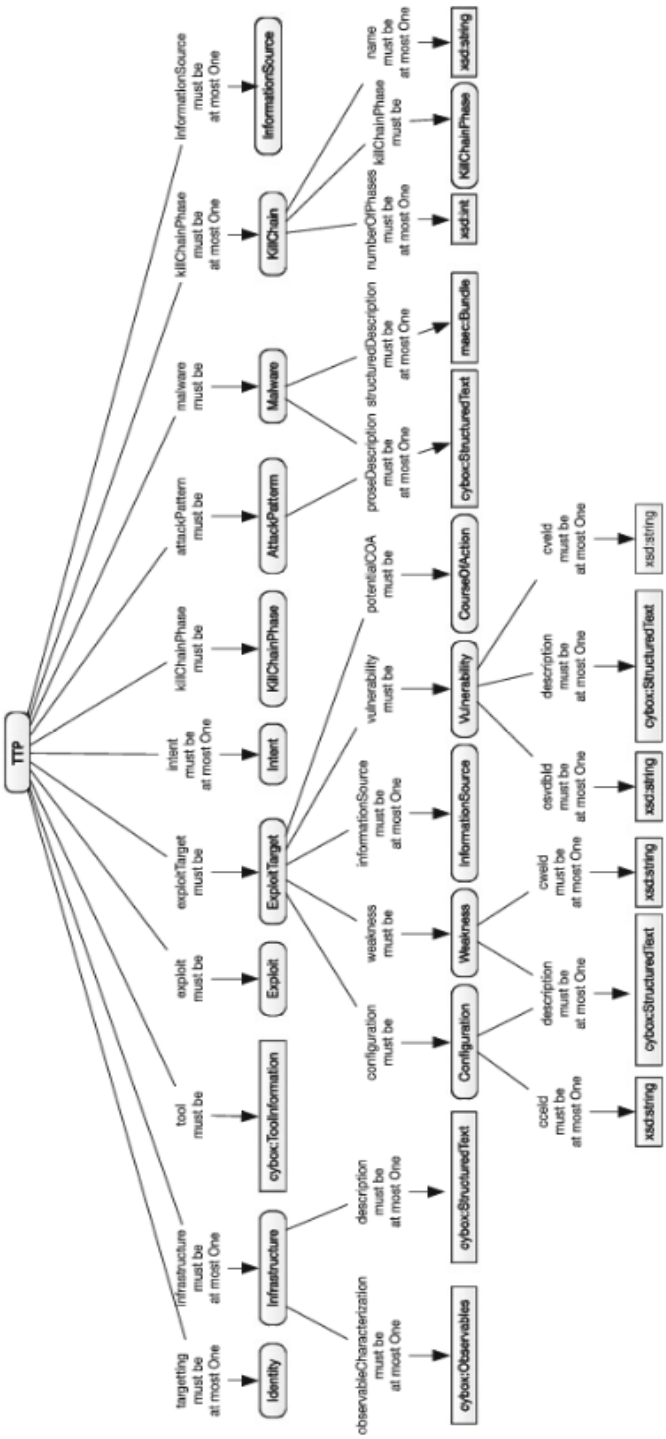


Figure 56 - High-level view of STIX ontology (Ulicny et al., 2014)

The work presented in this thesis is limited to ATT&CK, however these ontologies are included as a brief illustration of future directions for cyber threat information interoperability that are continuing to develop.

5.6 ATT&CK (Enterprise) Data Content High Level Summary

This section provides a summary of the Tactics (and the techniques associated with these tactics) that are described within the ATT&CK knowledge base. It also shows which techniques are used within the APT descriptions included in the knowledge base.

5.6.1 Basic Overview of Tactics

We have the following summary of ATT&CK Enterprise

ID	Name	Description
TA0043	Reconnaissance	The adversary is trying to gather information they can use to plan future operations.
TA0042	Resource Development	The adversary is trying to establish resources they can use to support operations.
TA0001	Initial Access	The adversary is trying to get into your network.
TA0002	Execution	The adversary is trying to run malicious code.
TA0003	Persistence	The adversary is trying to maintain their foothold.
TA0004	Privilege Escalation	The adversary is trying to gain higher-level permissions.
TA0005	Defense Evasion	The adversary is trying to avoid being detected.
TA0006	Credential Access	The adversary is trying to steal account names and passwords.
TA0007	Discovery	The adversary is trying to figure out your environment.
TA0008	Lateral Movement	The adversary is trying to move through your environment.
TA0009	Collection	The adversary is trying to gather data of interest to their goal.
TA0011	Command and Control	The adversary is trying to communicate with compromised systems to control them.
TA0010	Exfiltration	The adversary is trying to steal data.
TA0040	Impact	The adversary is trying to manipulate, interrupt, or destroy your systems and data.

Table 5 - ATT&CK Tactic Summary

There are 133 APTs (at current time) documented within ATT&CK. Below is a summary of Tactics used by all groups. It shows how many groups have intelligence relating to the use of each Tactic.

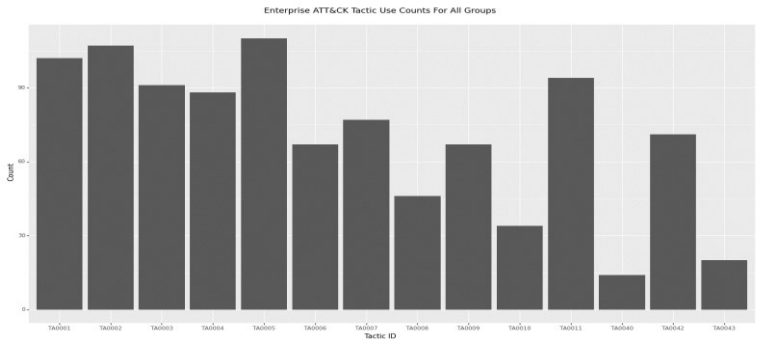


Figure 57 - Group Tactic Use Counts

As can be seen the data includes much less intelligence about the Reconnaissance (TA0043) and Impact (TA0040) 'phases' in the attack reports analysed.

A summary of different numbers of techniques used for each of the Tactics within Enterprise is shown below. This includes both main the technique levels and the associated sub-techniques.

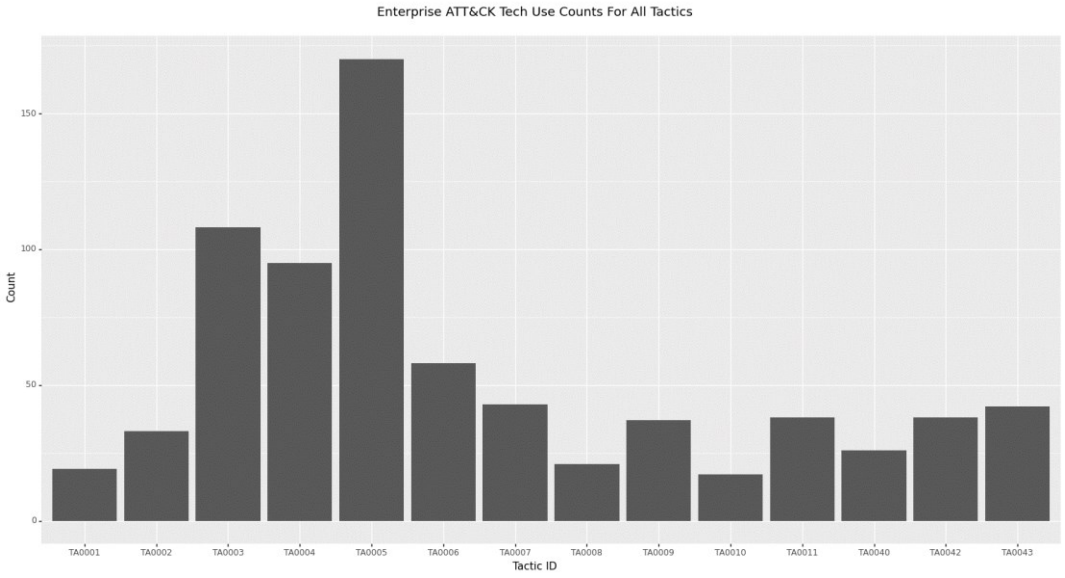


Figure 58 - Use of Techniques for Each Tactic

Note that most detail of intelligence is within TA0005 - Defense Evasion, TA0003 – Persistence and TA0004 – Privilege Escalation. Analysis leading to the definition of techniques across all tactics continues and this indicates that these areas have (to date) received the most attention.

Another view based only main techniques (ignoring sub-techniques) is shown below. This gives a more balanced view across the tactics and highlights where work on the definition of sub-techniques has been focussed. Now most detail of intelligence remains within TA0005 - Defense Evasion, but now TA0007 – Discovery is also seen as a significant source of intelligence

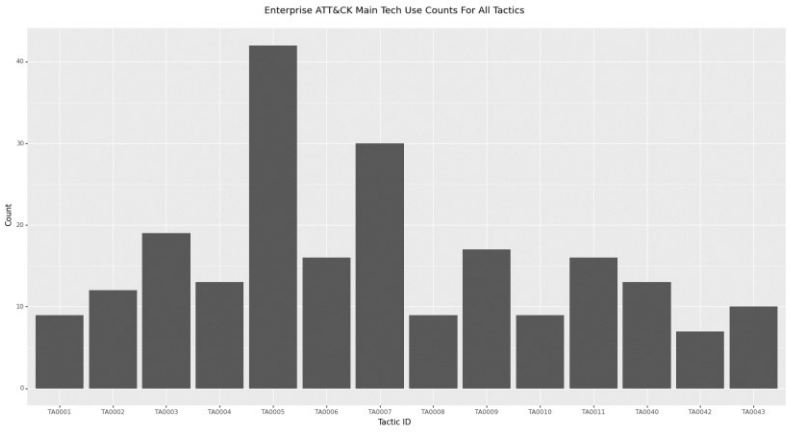


Figure 59 - Use of Main Techniques for Each Tactic

A summary of technique use across groups is shown below

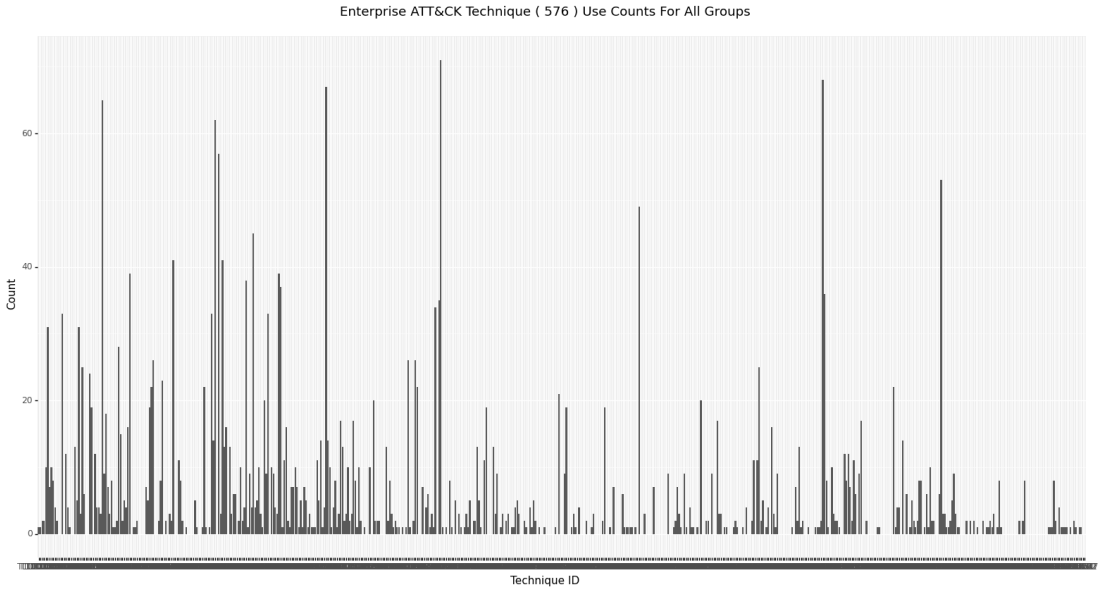


Figure 60 - ATT&CK Technique Use Across APTs

As there are many techniques, it easier to see specific use by breaking down by tactics. This does also show that not all techniques are represented in the current APT descriptions (out of a total of 576 techniques 372 are reported as used by the APTs included in the knowledge base). In the histograms presented below, the column *Tech* provides a summation of how many APTs have been observed using that individual *Tech*. The column ParentTech”TOT” provides a summation of all observations (parent and sub technique) for that parent technique.

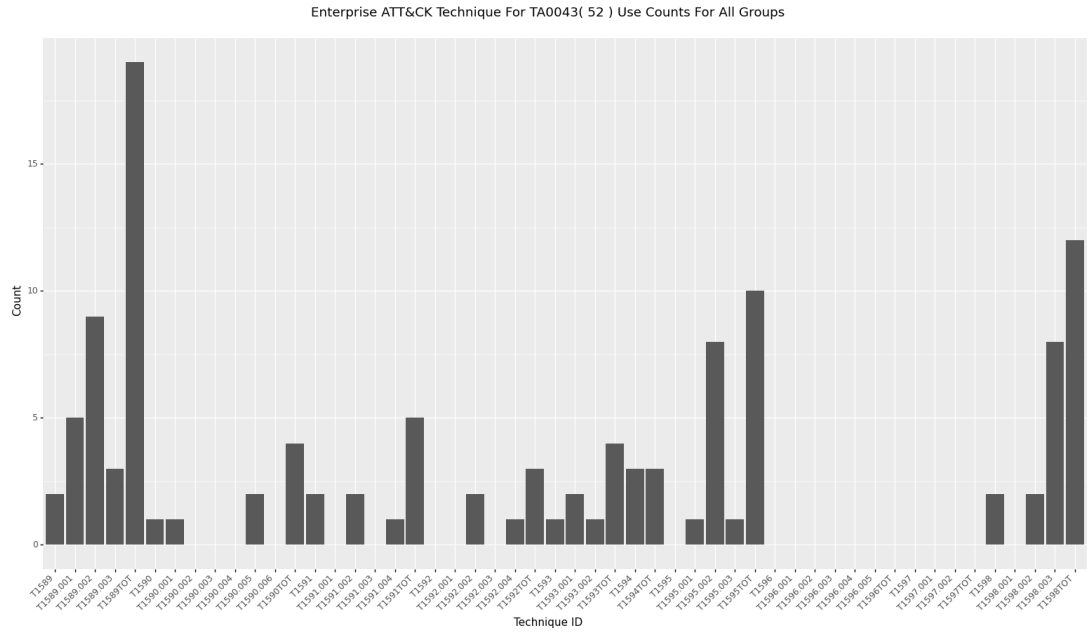


Figure 61 - ATT&CK TA0043 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the **Tactic TA0043 : Reconnaissance**. There are 52 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1590.002 : Gather Victim Network Information : DNS** and **T1592 : Gather Victim Host Information**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1590.002** with no documented use there are examples of APTs using the parent technique **T1590 : Gather Victim Network Information**, but conversely there are examples of sub-techniques observed (such as **T1592.002 Gather Victim Host Information : Software**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1589.002 : Gather Victim Identity Information : Email Addresses**. The most commonly observed parent technique is **T1589 : Gather Victim Identity Information**. The second most common is **T1598 : Phishing for Information**.

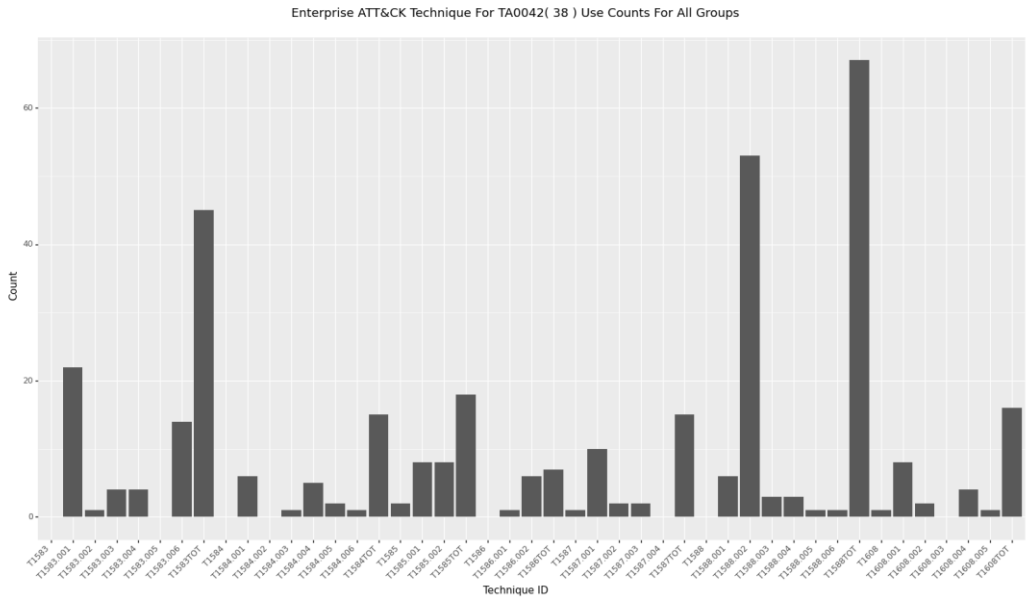


Figure 62 - ATT&CK TA0042 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the **Tactic TA0043 : Resource Development**. There are 38 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1583.005 : Acquire Infrastructure : Botnet** and **T1586 : Compromise Accounts**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1585.001 : Establish Accounts : Social Media Accounts** with no documented use there are examples of APTs using the parent technique **T1585 : Establish Accounts**, but conversely there are examples of sub-techniques observed (such as **T1583.001 : Acquire Infrastructure : Domains**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1588.002 : Obtain Capabilities : Tool**. The most commonly observed parent technique is **T1588 : Obtain Capabilities**. The second most common is **T1583 : Acquire Infrastructure**.

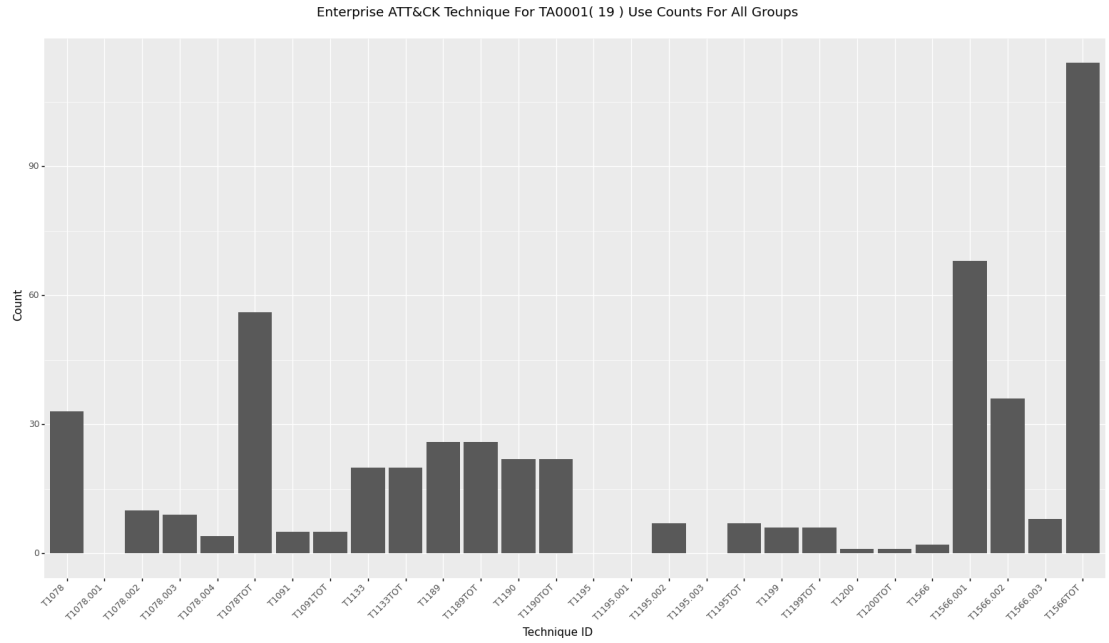


Figure 63 - ATT&CK TA0001 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the **Tactic TA0001 : Initial Access**. There are 19 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1078.001 : Valid Accounts : Default Accounts** and **T1195.001 : Supply Chain Compromise : Compromise Software Dependencies and Development Tools**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1078.001 : Valid Accounts : Default Accounts**, with no documented use, there are examples of APTs using the parent technique **T1078 : Valid Accounts**, but conversely there are examples of sub-techniques observed (such as **T1195.002 : Supply Chain Compromise : Compromise Software Supply Chain**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1566.001 : Phishing : Spearphishing Attachment**. The most commonly observed parent technique is **T1566 : Phishing**. The second most common is **T1078 : Valid Accounts**.

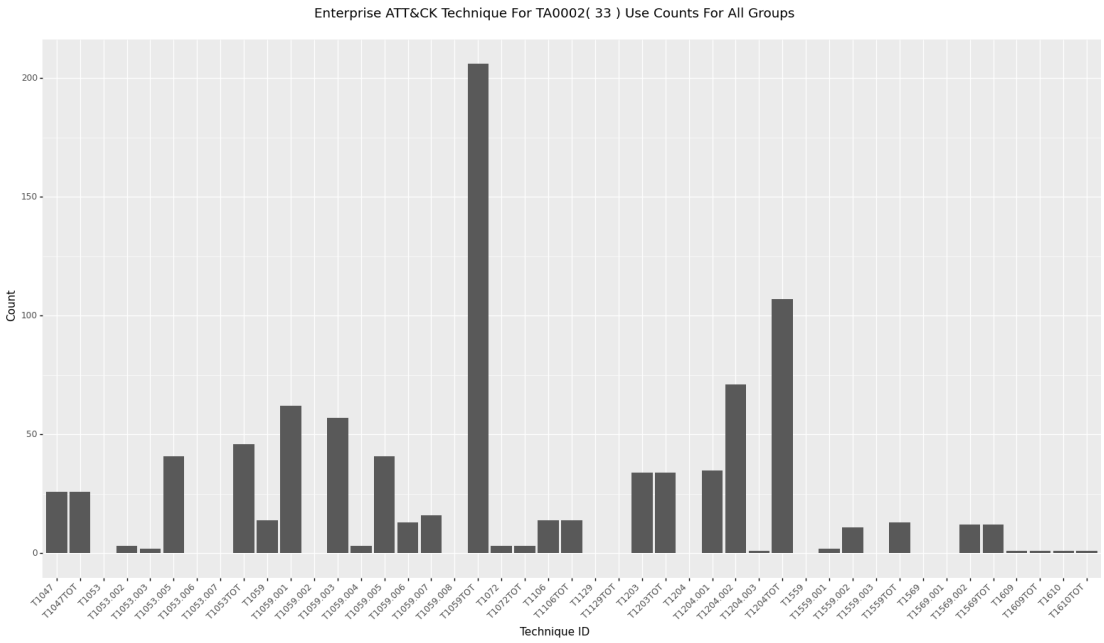


Figure 64 - ATT&CK TA0002 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0002 : Execution**. There are 33 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1059.008 : Command and Scripting Interpreter : Network Device CLI** and **T1559.003 : Inter-Process Communication : XPC Services**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1059.008 : Command and Scripting Interpreter : Network Device CLI**, with no documented use, there are examples of APTs using the parent technique **T1059 : Command and Scripting Interpreter**, but conversely there are examples of sub-techniques observed (such as **T1053.002 : Scheduled Task/Job : At**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1204.002 : User Execution : Malicious File**. The most commonly observed parent technique is **T1059 : Command and Scripting Interpreter**. The second most common is **T1204 : User Execution**.

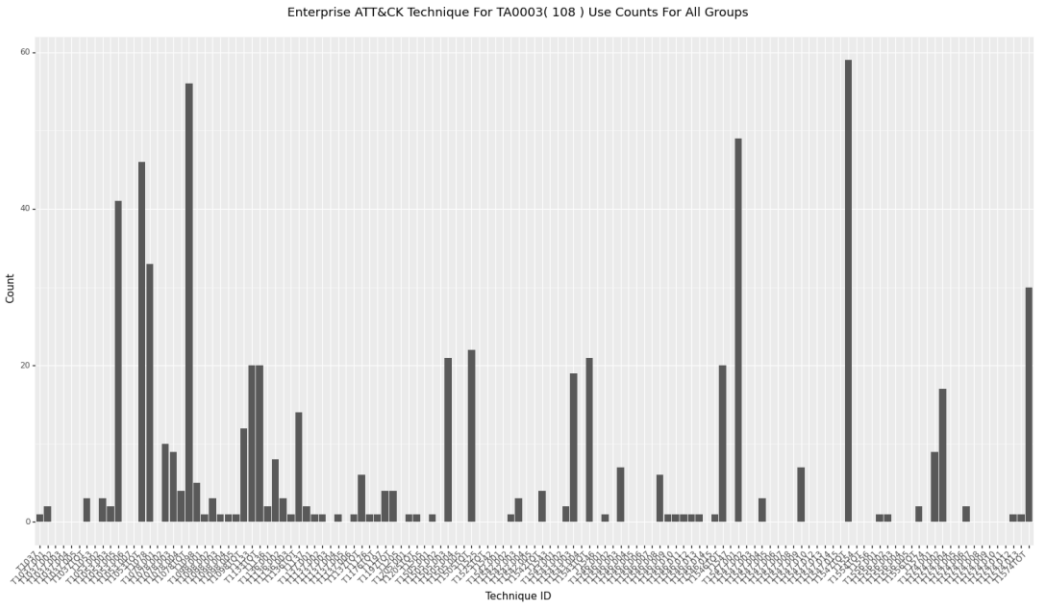


Figure 65 - ATT&CK TA0003 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0004 : Privilege Escalation**. There are 108 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1078.001 : Valid Accounts : Default Accounts** and **T1505.005 : Server Software Component : Terminal Services DLL**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1078.001 : Valid Accounts:Default Accounts**, with no documented use, there are examples of APTs using the parent technique **T1078 : Valid Accounts**, but conversely there are examples of sub-techniques observed (such as **T1205.001 : Traffic Signaling : Port Knocking**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1547.001 : Boot or Logon Autostart Execution : Registry Run Keys / Startup Folder**. The most commonly observed parent technique is **T1547 : Boot or Logon Autostart Execution**. The second most common is **T1078 : Valid Accounts**.

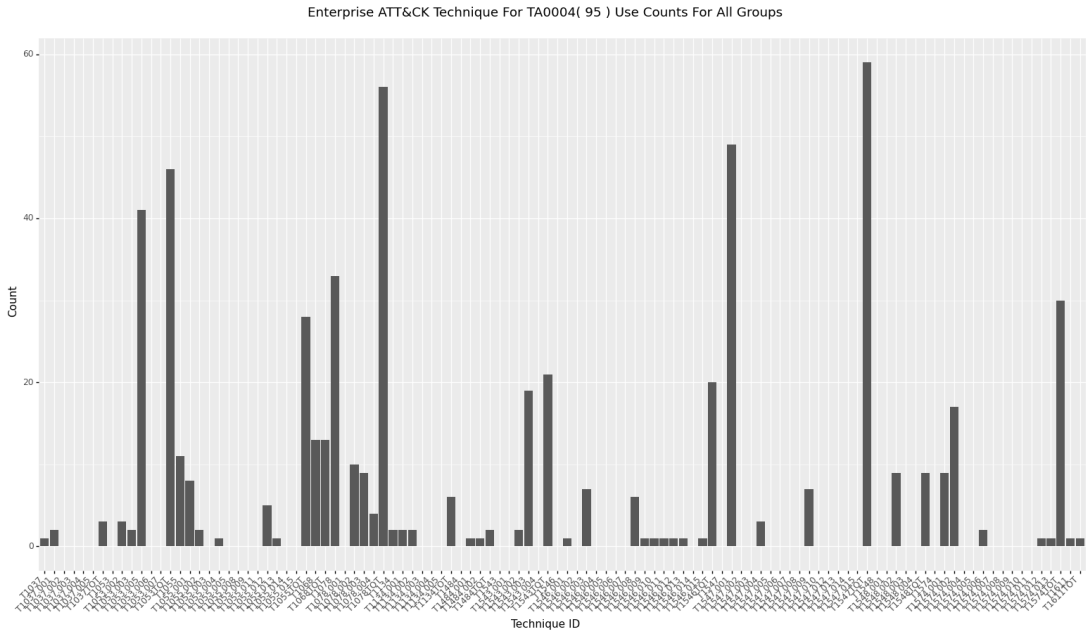


Figure 66 - ATT&CK TA0004 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0004 : Persistence**. There are 95 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1055.015 : Process Injection : ListPlanting** and **T1547.015 : Boot or Logon Autostart Execution : Login Items**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1055.015 : Process Injection : ListPlanting**, with no documented use, there are examples of APTs using the parent technique **T1055 : Process Injection**, but conversely there are examples of sub-techniques observed (such as **T1574.013 : Hijack Execution Flow : KernelCallbackTable**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1547.001 : Boot or Logon Autostart Execution : Registry Run Keys / Startup Folder**. The most commonly observed parent technique is **T1547 : Boot or Logon Autostart Execution**. The second most common is **T1078 : Valid Accounts**.

Similarities between the key points here and the results above for TA0003 are due to the fact that some techniques can be associated with multiple Tactics. In the APT descriptions provided no record is maintained of which Tactics are relevant to the techniques listed for the APTs.

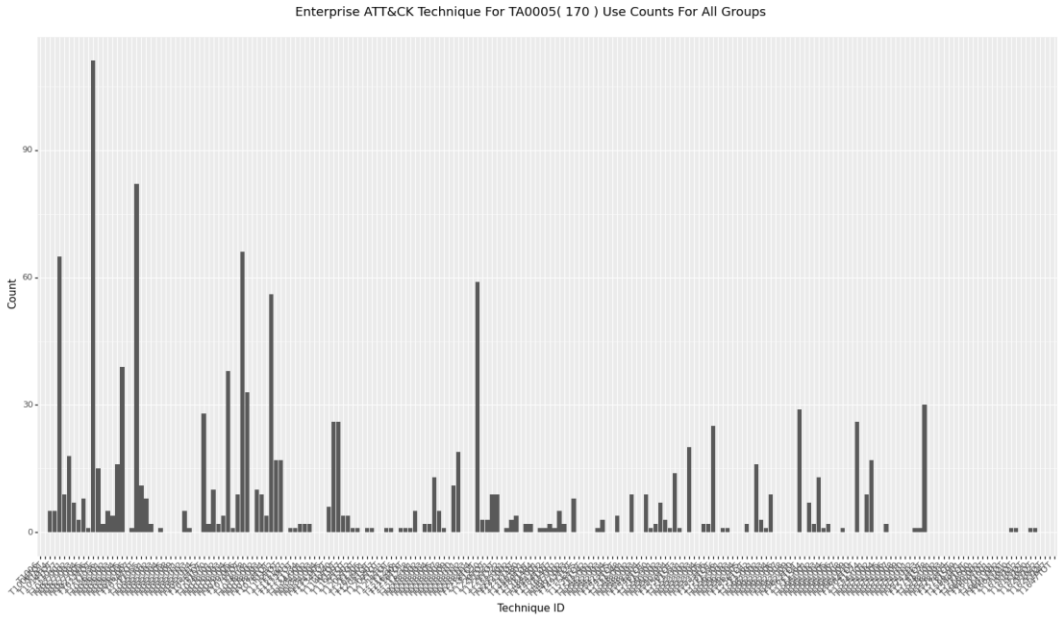


Figure 67 - ATT&CK TA0005 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0005 : Defense Evasion**. There are 170 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1218.014 : System Binary Proxy Execution : MMC** and **T1556.005 : Modify Authentication Process : Reversible Encryption**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1218.014 : System Binary Proxy Execution : MMC**, with no documented use, there are examples of APTs using the parent technique **T1218 : System Binary Proxy Execution**, but conversely there are examples of sub-techniques observed (such as **T1574.013 : Hijack Execution Flow : KernelCallbackTable**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1036.005 : Masquerading : Match Legitimate Name or Location**. The most commonly observed parent technique is **T1027 : Obfuscated Files or Information**. The second most common is **T1036 : Masquerading**.

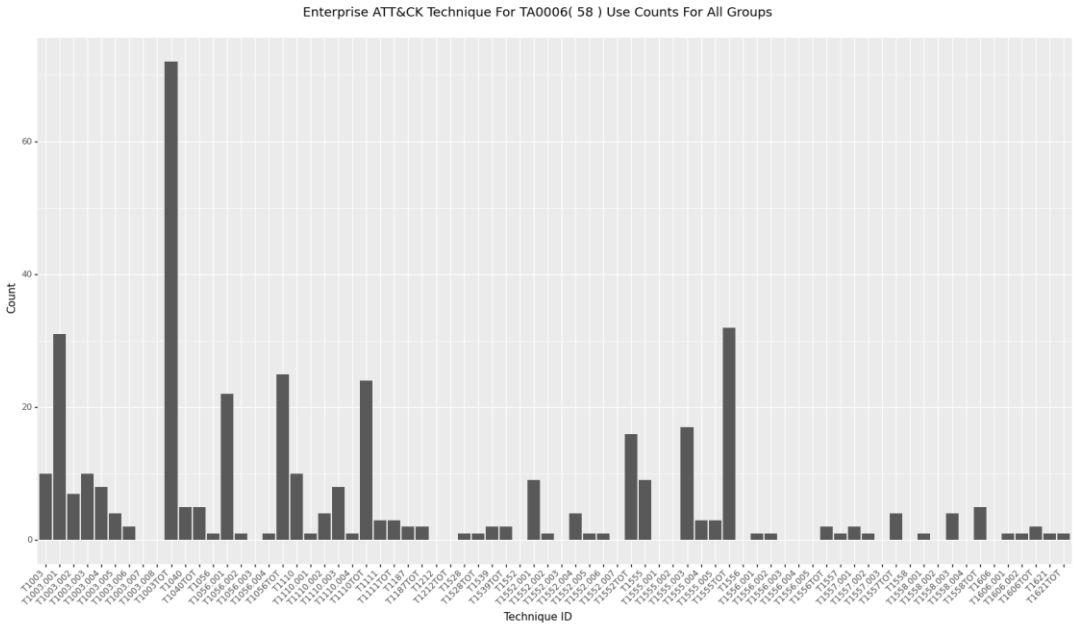


Figure 68 - ATT&CK TA0006 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0006 : Credential Access**. There are 58 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1555.002 : Credentials from Password Stores : Securityd Memory** and **T1552.007 : Unsecured Credentials : Container API**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1555.002 : Credentials from Password Stores : Securityd Memory**, with no documented use, there are examples of APTs using the parent technique **T1555 : Credentials from Password Stores**, but conversely there are examples of sub-techniques observed (such as **T1606.002 : Forge Web Credentials : SAML Tokens**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1003.001 : OS Credential Dumping : LSASS Memory**. The most commonly observed parent technique is **T1003 : OS Credential Dumping**. The second most common is **T1555 : Credentials from Password Stores**.

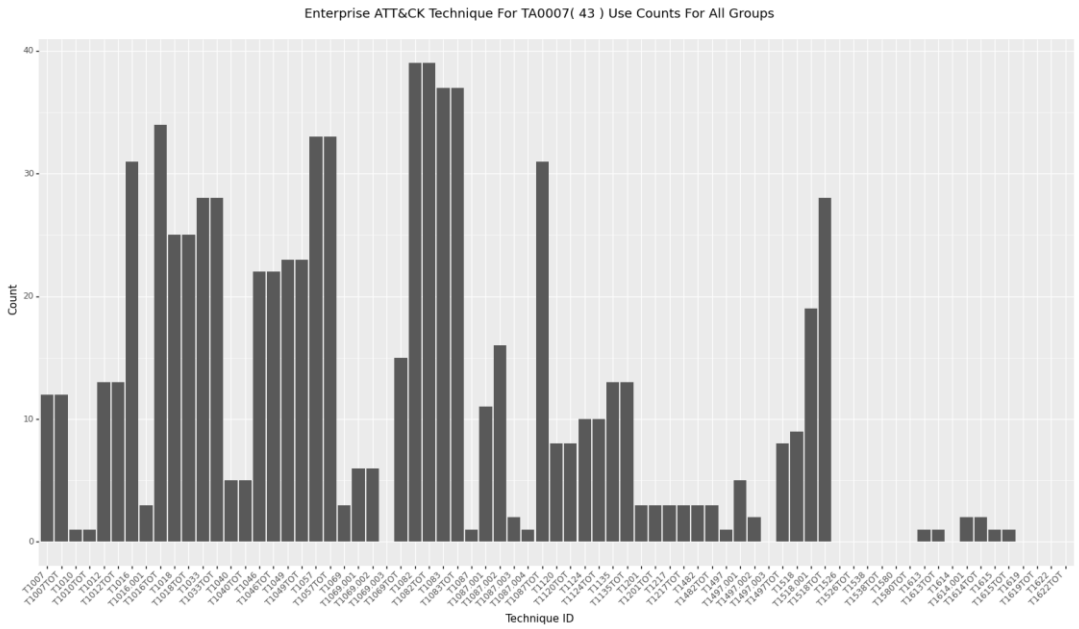


Figure 69 - ATT&CK TA0007 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0007 : Discovery**. There are 43 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1497.003 : Virtualization/Sandbox Evasion : Time Based Evasion** and **T1614.001 : System Location Discovery : System Language Discovery**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1497.003 : Virtualization/Sandbox Evasion : Time Based Evasion**, with no documented use, there are examples of APTs using the parent technique **T1497 : Virtualization/Sandbox Evasion**.

We can also see that the most commonly observed sub-technique is **T1518.001 : Software Discovery : Security Software Discovery**. The most commonly observed parent technique is **T1082 : System Information Discovery**. The second most common is **T1083 : File and Directory Discovery**.

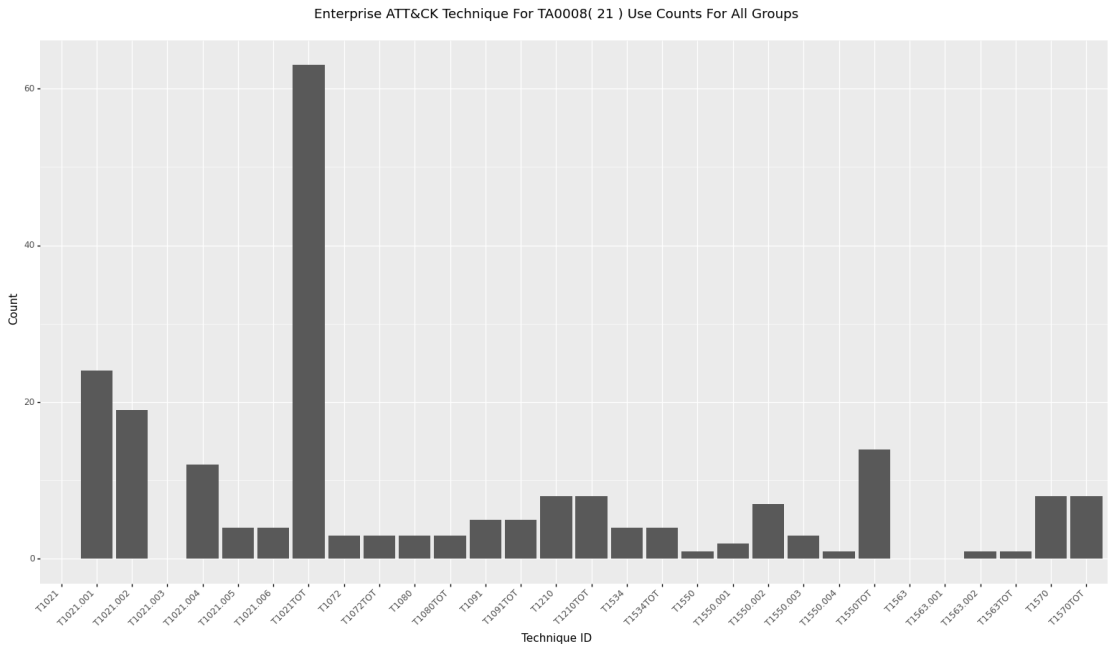


Figure 70 - ATT&CK TA0008 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0008 : Lateral Movement**. There are 21 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1563.001 : Remote Service Session Hijacking : SSH Hijacking** and **T1021.003 : Remote Services : Distributed Component Object Model**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

There are examples of sub-techniques observed (such as **T1563.002 : Remote Service Session Hijacking : RDP Hijacking**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1021.001 : Remote Services : Remote Desktop Protocol**. The most commonly observed parent technique is **T1021 : Remote Services**. The second most common is **T1550 : Use Alternate Authentication Material**.

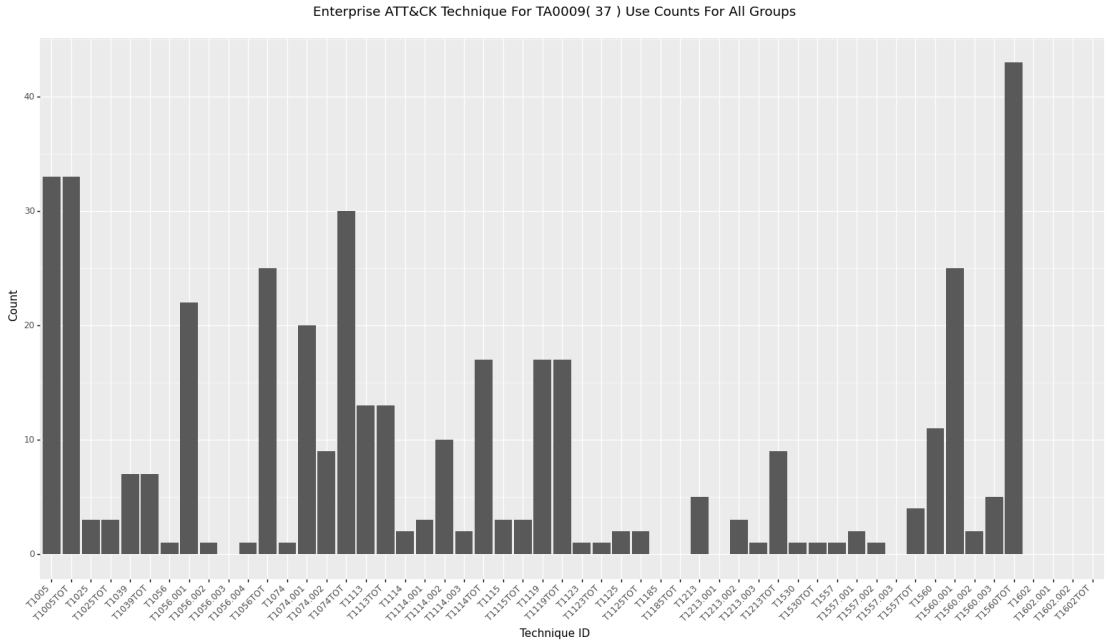


Figure 71 - ATT&CK TA0009 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0009 : Collection**. There are 37 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1557.003 : Adversary-in-the-Middle : DHCP Spoofing** and **T1602.002 : Data from Configuration Repository : Network Device Configuration Dump**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1557.003 : Adversary-in-the-Middle : DHCP Spoofing**, with no documented use, there are examples of APTs using the parent technique **T1555 : Credentials from Password Stores**.

We can also see that the most commonly observed sub-technique is **T1560.001 : Archive Collected Data : Archive via Utility**. The most commonly observed parent technique is **T1560 : Archive Collected Data**. The second most common is **T1005 : Data from Local System**.

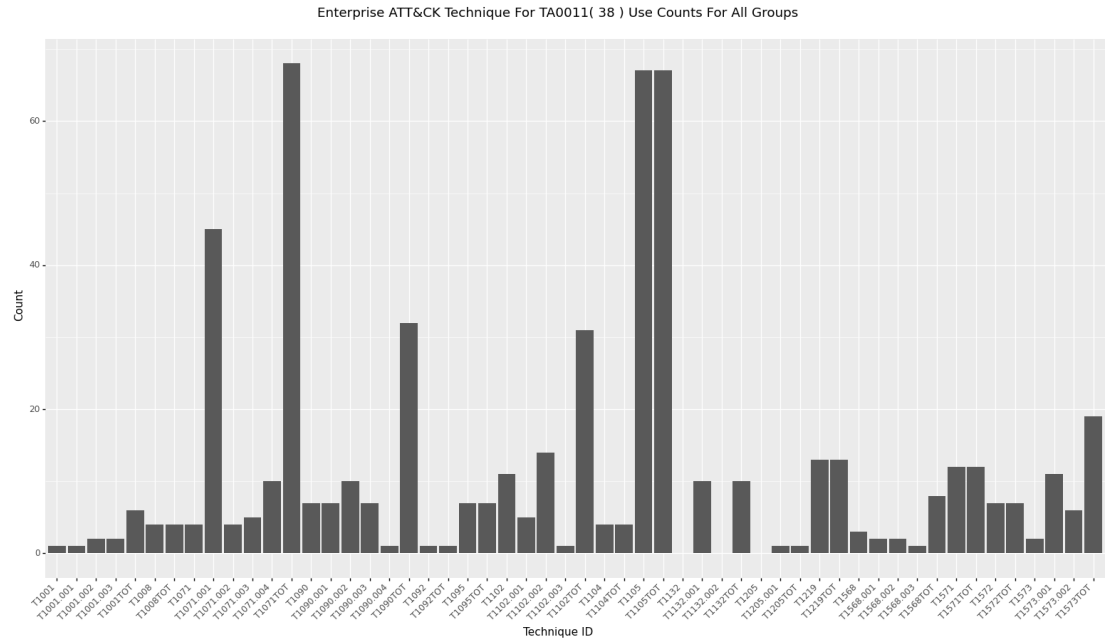


Figure 72 - - ATT&CK TA0011 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0011 : Command and Control**. There are 38 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1132.002 : Data Encoding : Non-Standard Encoding** and **T1205 : Traffic Signaling**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

There are examples of sub-techniques observed (such as **T1205.001 : Traffic Signaling : Port Knocking** and **T1132.001 : Data Encoding : Standard Encoding**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1071.001 : Application Layer Protocol : Web Protocols**. The most commonly observed parent technique is **T1071 : Application Layer Protocol**. The second most common is **T1105 : Ingress Tool Transfer**.

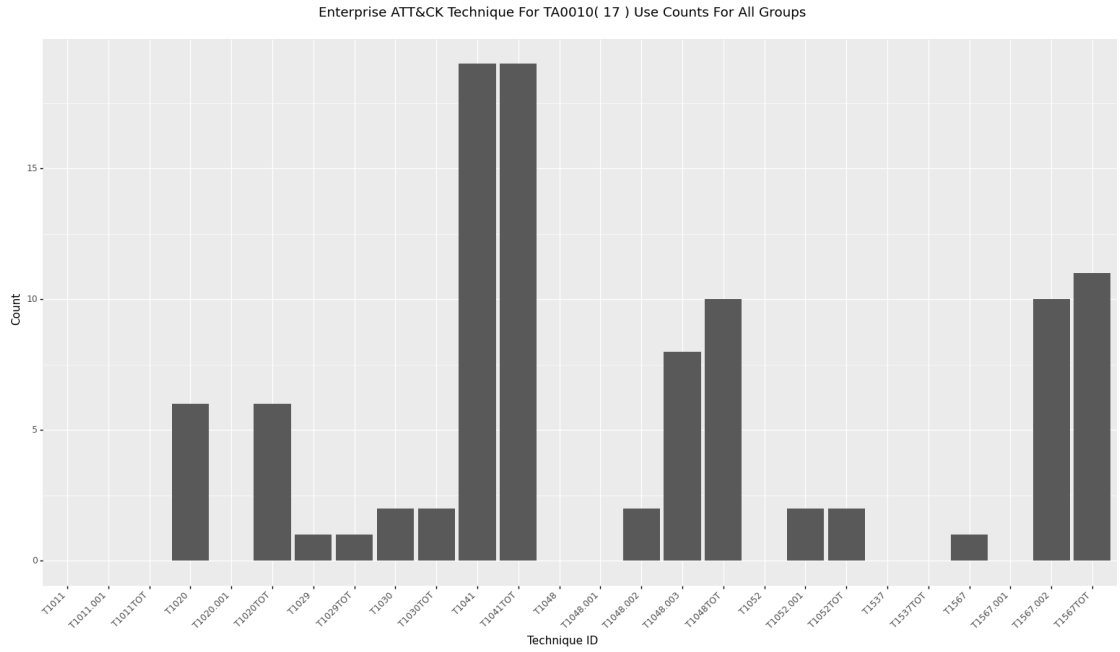


Figure 73 - ATT&CK TA0010 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0010 : Exfiltration**. There are 17 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1020.001 : Automated Exfiltration : Traffic Duplication** and **T1048.001 : Exfiltration Over Alternative Protocol : Exfiltration Over Symmetric Encrypted Non-C2 Protocol**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1020.001 : Automated Exfiltration : Traffic Duplication**, with no documented use, there are examples of APTs using the parent technique **T1020 : Automated Exfiltration**, but conversely there are examples of sub-techniques observed (such as **T1048.003 : Exfiltration Over Alternative Protocol : Exfiltration Over Unencrypted Non-C2 Protocol**) where use of the parent technique has not been recorded. This may be partially due to sub-techniques being introduced at a later date.

We can also see that the most commonly observed sub-technique is **T1567.002 : Exfiltration Over Web Service : Exfiltration to Cloud Storage**. The most commonly observed parent technique is **T1041 : Exfiltration Over C2 Channel**. The second most common is **T1567 : Exfiltration Over Web Service**.

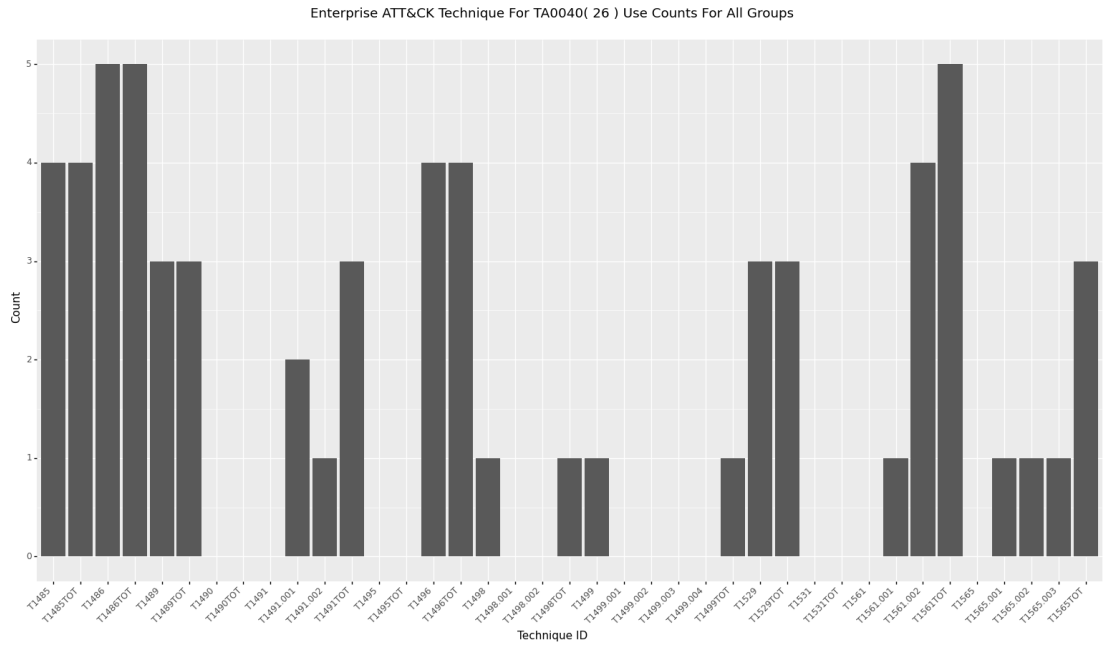


Figure 74 - ATT&CK TA0040 Technique Use

This histogram shows how many APTs are recorded as using each Technique associated with the Tactic **TA0040 : Impact**. There are 26 separate techniques and sub-techniques. Here we can see examples of Techniques (such as **T1498.002 : Network Denial of Service : Reflection Amplification** and **T1498.001 : Network Denial of Service : Direct Network Flood**) that are defined in the knowledge base but none of the APTs documented are noted as using these techniques.

For **T1498.002 : Network Denial of Service : Reflection Amplification**, with no documented use, there are examples of APTs using the parent technique **T1498 : Network Denial of Service**.

We can also see that the most commonly observed sub-technique is **T1561.002 : Disk Wipe : Disk Structure Wipe**. The most commonly observed parent technique is **T1561 : Disk Wipe**. The second most common is **T1486 : Data Encrypted for Impact**.

5.7 An Initial Attempt at Clustering Group ‘Fingerprints’

This section looks at whether clusters of the sets of tactics used by APTs can be discerned. There are many clustering approaches but in general they try to group data with similar characteristics. This was originally actioned to see if such an appropriate clustering could help provide additional intelligence to analysts who had observed ATT&CK tactic/technique use within a system (that is relate sets of observations to groups of patterns of techniques).

The results were not conclusive and this matched a small number of related studies (referenced below). The results here were included here as further descriptive insight into the characterisation of the ATT&CK data content.

By representing the use of the 14 Enterprise tactics by each group as binary vector (i.e. 1 for used and 0 if not) and building on work done in (R. Al-Shaer et al., 2020) we can look at whether there is clustering across the group tactic use.

Two examples of such binary vectors are given below (in the rows of this table).

	Tactics	**0043**	**0042**	**0001**	**0002**	**0003**	**0004**	**0005**	**0006**	**0007**	**0008**	**0009**	**0011**	**0010**	**0040**
Groups															
admin@338		0	0	1	1	0	0	1	0	1	0	0	0	0	0
Ajax Security Team		0	0	1	1	0	0	0	1	0	0	1	1	0	0

Table 6 - Example APT Tactic Use Vectors

As there are 133 APTs defined in the knowledge base we end up with 133 14 'bit' binary vectors.

5.7.1 'Clusterability' of the Data

One approach to looking at the clustering tendency ('clusterability') of a data set is using the Hopkins' statistic (Hopkins & Skelllam, 1954) (Lawson & C, 1990). The Hopkins statistic is known to be a fair estimator of randomness in a data set (Banerjee & Davé, 2004).

Given a set X of n data points (with D variables) in sample space U. The approach is summarised as follows from (Wright, 2022):-

- Sample at random a data point (without replacement (Wikipedia, 2023a)) from X. Calculate the distance (generalised beyond just Euclidean distance see (Banerjee & Davé, 2004) (Wikipedia, 2023a)) to its nearest neighbour in X.
- Generate one new point Uniformly distributed in the sample space U. Calculate the distance from this point to the nearest neighbour in X?
- Repeat the two steps above m times (where m << n, advice is around 0.1n)
- Calculate H = $\sum_{i=1}^{n}(y_i)^d / (\sum_{i=1}^{n}(x_i)^d + \prod_{i=1}^{n}(y_i)^d)$.

From (Banerjee & Davé, 2004) we also define hypotheses to be considered:

- H_0 : The distances from the randomly sampled points (from U) to the nearest points in X should, on average be the same as the nearest distances between a member of X and its nearest neighbour (i.e. X is randomly distributed).
- H_1 : The distances from the randomly sampled points (from U) to the nearest points in X should, on average be the greater than the nearest distances between a member of X and its nearest neighbour (i.e. X contains clusters).

From (Banerjee & Davé, 2004) (and others) we also note:

- H has a Beta distribution with parameters (m,m) and always lies between 0 and 1 (Wright, 2022)
- H > 0.5 (almost 1.0) for well-defined clustered data
- H << 0.5 for regularly spaced data (neither clustered or random)
- H ≈ 0.5 for random data

In (Wright, 2022) a protocol is provided for using the Hopkins statistic. Elements of this protocol are discussed below

It suggested that the number of data points (n) in X is greater than 100, for this work we have 133 data points. It is also suggested that the number of uniform samples (m) from U is equal to 0.1 n. Investigation of the APTs showed a 'lowest' value of [0]*14 (a 14 bit binary vector of 0s) and the 'highest' value of [1]*14. So for this test uniform 0.1*133 (the number of APT) 14-bit vectors where uniformly sampled in this range.

The Hamming distance is a very simple measure it simply adds up the number of tactics two groups have in common. Using it does not include any additional knowledge about relationships between tactic use or how 'near' one tactic may be to another. By using the Hamming distance we have a distance measure between binary vector data points (see also example in (R. Al-Shaer et al., 2020)). The Hamming distance was the measure used in this sample space (see also code at (Maidens, 2023)).

However, in this protocol there are a couple of areas that suggest the use of the Hopkins statistic in this setting could be treated with caution. There are questions about the spatial randomness of the data points (for example, tactics like Initial Access are very commonly observed in APTs perhaps creating correlation between the data points) and also the dimension of the data is 14 which is much greater than 2 (the suggested safe limit before 'edge-effects' may occur, although in this case the boundaries of the sample space are defined by the vector types).

As also well described in (Wright, 2022) there have been a number of developments and interpretations of the definition of the Hopkins statistic (indeed (R. Al-Shaer et al., 2020) seem to use a different interpretation of the H formula (removing the power of d). The version presented above seems to be consistent across recent literature. Implementations of Hoskins available in Python and R seem to vary on how they implement the statistic and are quite 'opaque' about exact implementation details.

Despite the notes above a Hoskin's statistic for 133 14 'bit' binary vectors was calculated. This was achieved using bespoke code implementing Hoskin's (this can be found at (Maidens, 2023)).

Multiple runs (to average over random sample spaces) provided a H value of > 0.9. This value suggests clustering in the data. This result is simply included for completeness. Accuracy and interpretation subject to the concerns noted above and is simply used illustratively. This investigation was only used to motivate further consideration of clustering, shown below.

5.7.2 Creating Data Clusters

As described in the Introduction above the input data is 133 14-bit binary vectors (representing observation of tactic use by each of the APTs).

Similar to the discussion in (R. Al-Shaer et al., 2020) a hierarchical clustering approach is investigated here.

Hierarchical clustering is a type of clustering algorithm that seeks to define potential clusters. It breaks the data down into 1 to n clusters, where n is the number of observations in the data set. There are two approaches: Divisive (top-down) and Agglomerative (bottom-up). It is a flexible approach as it can be applied to any data where a distance can be calculated between observations.

Agglomerative is a bottom-up approach and starts with each observation as a separate cluster. Pairs of closest clusters are merged hierarchically until the data is represented as a single cluster.

Divisive is a top-down approach and starts with all observations in a single cluster. Clusters are then split into two hierarchically until the data is represented as a single cluster.

In both approaches 'closeness' is understood through a distance measure and recalculated as the new hierarchy of clusters are resolved. For the binary data type here Hamming distance was used as the appropriate distance measure (refinements discussed in (R. Al-Shaer et al., 2020) were not used here).

As well as a distance measure hierarchical clustering also uses a Linkage Method. This is used to measure the dissimilarity between sets of observations (as they are created). Through experimentation the ward method (this method attempts to minimise the variance within clusters) was selected.

The hierarchical clustering approaches described above result in a tree like structure (a dendrogram) representing the clustering structure within the data.

This approach was selected as this section is illustrative and the hierarchical approach is straightforward to implement. Either hierarchical approach was appropriate to the objective here, but Agglomerative clustering was chosen in this case. This was accessible with the required distance measure and linkage methods through standard R functions (see (Maidens, 2023)).

As a first step we can visualise the distance matrix for the input data. This heat map visually suggests three distinct clusters with a less distinct fourth (possibly itself made of two sub clusters).

From R (via the factoextra package)

```
DistanceMatrix=rdist(BinVecData, metric = "hamming")
fviz_dist(DistanceMatrix)
```

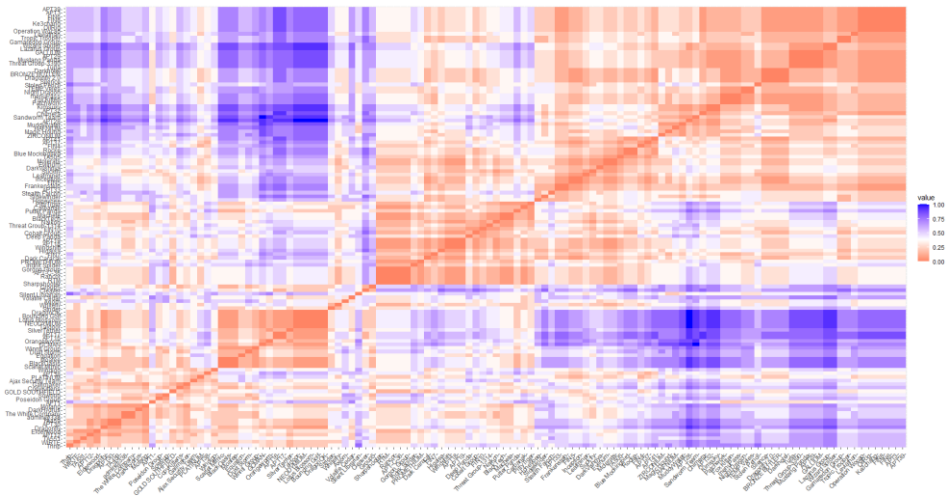


Figure 75 - Heat Map APT Distance Measures

The resulting dendogram is presented below (with a visual suggestion of potentially 4 major clusters). It should also be said that investigating unsupervised approaches such as using the R utility nbclust() (Rdocumentation.org, 2023) (which runs thirty different indices for determining best number of clusters) suggested a value of 2 major clusters. Although a silhouette() (Wikipedia, 2023c) type approach also visually suggests 4 clusters. I have not detailed these two additional

approaches further here (and just included for commentary) as the cluster results are intended for illustration only, as noted below.

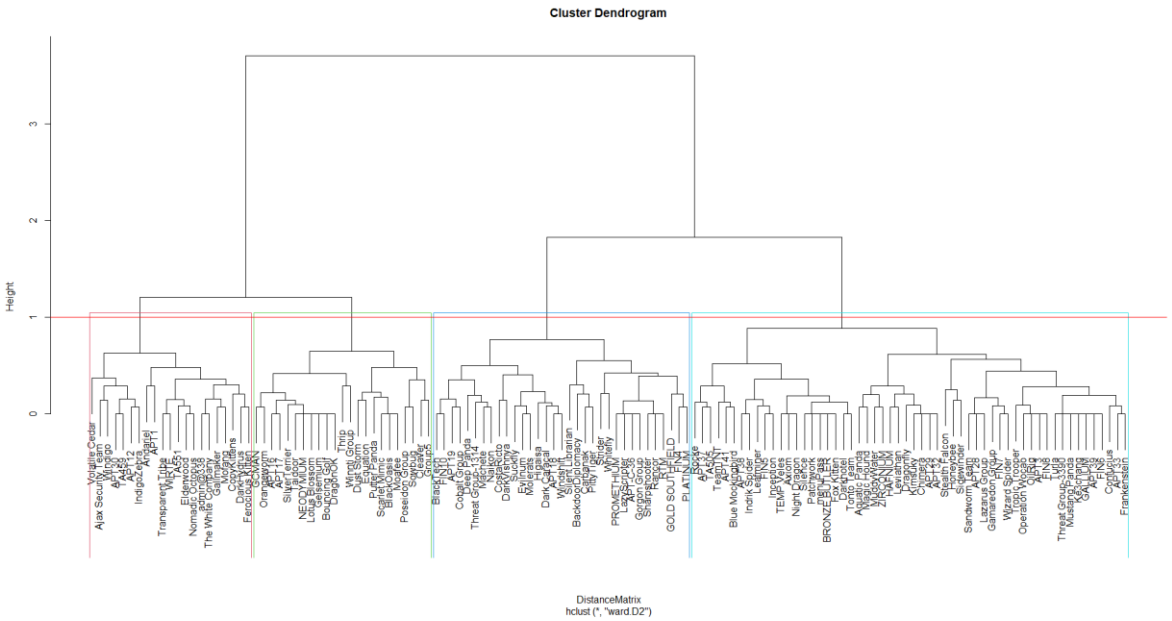


Figure 76 - APT Cluster Dendrogram 1

The clusters shown illustrated above are visualised as shown below. The Tactics are shown on the x-axis that is "TA0043", "TA0042", "TA0001", "TA0002", "TA0003", "TA0004", "TA0005", "TA0006", "TA0007", "TA0008", "TA0009", "TA0011", "TA0010" and "TA0040". The groups are shown on the y-axis. Each row shows the binary vector for each group. The light squares represent a zero (no observed tactic use) and the dark squares represent a 1 (observed tactic use).

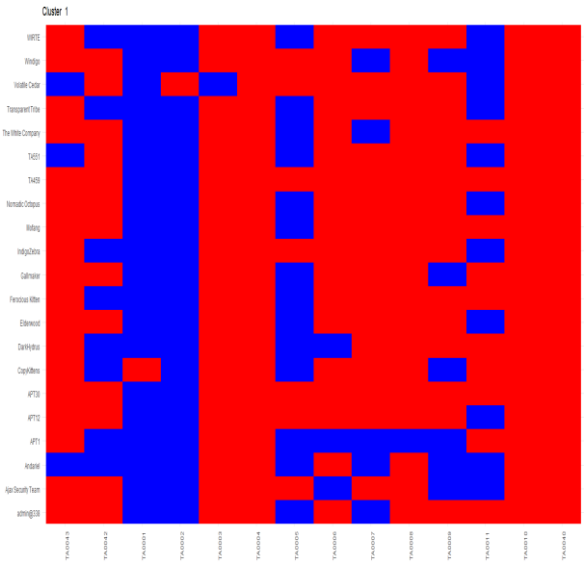


Figure 77 - APT Tactic observations Cluster 1

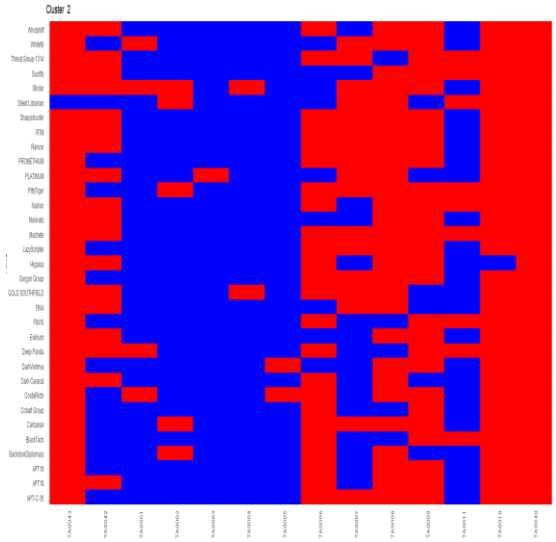


Figure 78 - APT Tactic observations Cluster 2

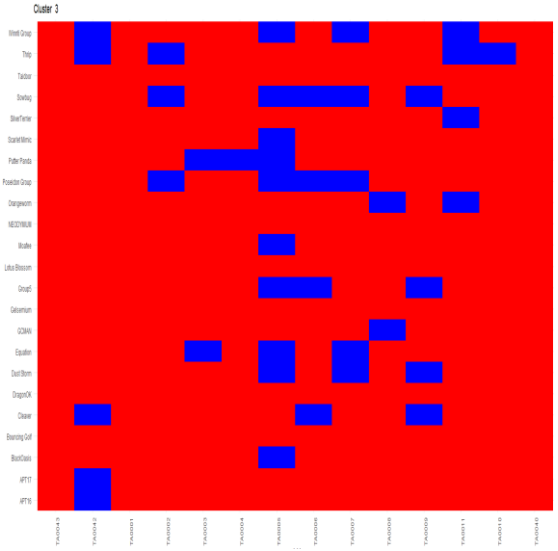


Figure 79 - APT Tactic observations Cluster 3

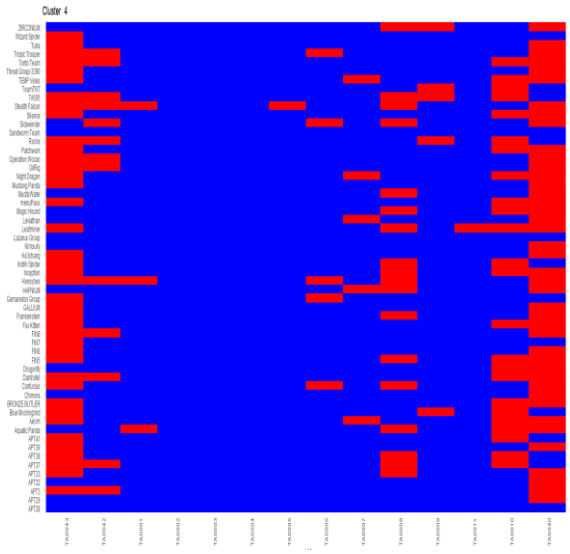


Figure 80 - APT Tactic observations Cluster 4

Although a simple visual approach has been used here. This does suggest that actually the clustering is really grouping the APTs by the amount of intelligence available.

- Cluster 4 (the largest cluster) represents groups where knowledge of use is broadly across most/all Tactics
- Cluster 3 represents groups where there is little knowledge of use Tactics
- Cluster 1 represents groups where knowledge is mostly available for TA0001 – Initial Access, TA0002 – Execution and TA0005 – Defense Evasion (and to a lesser extent TA0011 – Command & Control).
- Cluster 2 represents groups where knowledge is mostly available for TA0001 – Initial Access, TA0002 – Execution, TA0003 – Persistence, TA0004 – Privilege Escalation and TA0005 – Defense Evasion and TA0011 – Command & Control.
- All groups have weaker coverage of TA0040 – Impact ,TA0042 – Resource Development, TA0043 – Reconnaissance suggesting that the ATT&CK focus is currently predominantly (and understandably) technical.

We do not know whether a lack of observation of a tactic really indicates a characteristic of the APT, a lack of intelligence available generally or a failure to curate available intelligence to update the APT description. For instance, for APTs in Cluster 3 we have Neodymium, Taidoor, and Gelsemium with no tactics observed at all (these APTs do have records of software used). For Taidoor and Gelsemium this may also reflect early identification of groups while analysts finalise decisions on whether the behaviours observed are truly associated with a group.

Another approach is discussed in (Wang et al., 2020), “Clustering using a similarity measure approach based on semantic analysis of adversary behaviours”. However not enough intelligence is available to pursue this approach, which consider Goals, Behaviours and Capabilities.

5.8 Conclusion

MITRE ATT&CK is becoming a major standard for describing the TTPs used in cyber-attacks. As well as the core knowledge base, there are now a number of associated developments, including approaches to associating alerts with standard techniques.

The collated knowledge on various APTs does provide some additional insight however this is not very fine grained. Further investigation into the clustering described above reveals that the main clusters really highlight the varying ‘coverage’ of intelligence provided for the APTs.

The key limitation in this model as briefly discussed in (Spring & Al-shaer, 2020) (Summary slide 15) is around the lack of temporal intelligence included with the current version. Related discussion is also included in (E. Al-Shaer & Chu, 2017) and the STIXChecker development

Chapter 6 Building a Model

6.1 Introduction

The intention of this thesis is to show how a knowledge base of attacks (described as a sequence of ATT&CK Tactics and Techniques) can provide an additional source of cyber threat intelligence to complement the existing MITRE ATT&CK Knowledge Base.

To demonstrate this, fragments of cyber-attacks described in open-source intelligence are taken and described as sequences of ATT&CK techniques.

In this section a model for describing these sequences is built and explained.

Examples of how these sequences may be used to provide input to a user's potential cyber situational awareness are then given in the next Chapter Results.

An overview of how this is considered is as follows:

1. A summary of how the MITRE ATT&CK data was prepared for this work (see also Loading the Base Data into a Queryable Model).
2. Creation of a simple motivating example to show current limitations of expressing APT behaviours as a list of techniques without sequence (see also Motivating Example).
3. Development of initial model to record an attack sequence populated with a small number of examples (see also An Initial Attempt at Recording Attacks as Sequences).
4. Development of a new model based on observations from the initial model above (see also A New Attack Model)
    a. A meta model
    b. A sequence model
5. The old and new model are compared (see also Comparing the Initial and New Attack Model)
6. The new model is further populated with a representative data set to ensure coverage of data and relationship types (see Loading a Representative Data Set)
    a. Attack sequences from 8 attacks have been used to exercise the model initially.
    b. The data from these 8 attacks is reviewed against coverage of MITRE Tactics (see An Analysis Against ATT&CK Tactics/Techniques) and Unified Kill Chain steps (see An Analysis Against Unified Kill Chain).
    c. The data from 7 more attack are added to complete coverage (see Adding Further Attacks to Complete Tactic and UKC Phase Coverage).

d. 9 more attacks are added to include additional APTs commonly referenced cyber-attack studies (see Adding Further Attacks to Demonstrate Additional APT Groups).
e. This completes the set of attacks created to be used in the example applications given in Results. A summary of this set of 26 attacks is given in Test Set Summary.

6.2 Loading the Base Data into a Queryable Model

This is the next step taken to define an attack sequence model as outlined in Introduction.

The base ATT&CK data has been loaded from a current MITRE repository.

This has been loaded using python code developed for this work (Maidens, 2023).

It is initially loaded into Python Panda Dataframes that are then persisted as CSV files that can also be accessed using Excel. These Dataframes have been designed to act as tables in a Relational Database (and indeed these can also be used to construct a simple lightweight SQLite database).

A number of support functions were also developed to allow easy access to common queries, and these have been used to create 'test rigs' to run the investigations reported on below.

These are all developed as Python objects. In this way the ATT&CK data is made more easily accessible from 'application' code units used in this work.

6.3 Motivating Example

6.3.1 Introduction

This is the next step taken to define an attack sequence model as outlined in Introduction.

Here an example is described to show how just listing the techniques used by an APT in an attack can create limitations in the interpretation of observed techniques in a system.

6.3.2 A Motivating Simple Initial Example

Based on a report for APT group admin@338 (Mandiant, n.d.)

The core elements of the attack are summarised here for reference.

August 2015

Spearphishing - Hong Kong based orgs (**T1566.001 Phishing: Spearphishing Attachment**)

Three MS Word .doc files (**CVE-2012-0158**) (**T1203 Exploitation for Client Execution**) (**T1204.002 User Execution: Malicious File**)

Attachment – **LOWBALL malware**

Indicators of compromise for malware provided

This backdoor uses Dropbox cloud storage service to act as C&C server

Dropbox API with hardcoded bearer access token

Upload/download/execute files (**T1102.002 Web Service: Bidirectional Communication**)

HTTPS port 443 (**T1071.001 Application Layer Protocol: Web Protocols**)

Downloads WmiApCom.bat (to start WmiApCom and download new version of LOWBALL) (**T1105 Ingress Tool Transfer**)

Threatgroup monitors C&C

Create .bat file and execute on target (**T1059.003 Command and Scripting Interpreter: Windows Command Shell**)

We observed the threat group issue the following commands:

```
@echo off
dir c:\ >> %temp%\download ( T1083 File and Directory Discovery )
 ipconfig /all >> %temp%\download
 net user >> %temp%\download ( T1016 System Network Configuration Discovery )
 net user /domain >> %temp%\download
 ver >> %temp%\download ( T1082 System Information Discovery )
 del %0
@echo off
 dir "c:\Documents and Settings" >> %temp%\download
 dir "c:\Program Files\
 " >> %temp%\download
 net start >> %temp%\download ( T1007 System Service Discovery )
 net localgroup administrator >> %temp%\download ( T1069.001 Permission Groups Discovery: Local Groups )
 netstat -ano >> %temp%\download ( T1049 System Network Connections Discovery )
```

Results stored in file and uploaded to C&C server

These commands allow the threat group to gain information about the compromised computer and the network to which it belongs.

Using this information, they can decide to explore further or instruct the compromised computer to download additional malware.

Download second stage malware **BUBBLEWRAP** (Backdoor.APT.FakeWinHTTPHelper)

BUBBLEWRAP is a full-featured backdoor that is set to run when the system boots, and can communicate using HTTP, HTTPS, or a SOCKS proxy. This backdoor collects system information, including the operating system version and hostname, and includes functionality to check, upload, and register plugins that can further enhance its capabilities.

```
@echo off
 ren "%temp%\upload" audiodg.exe ( T1036.005 Masquerading: Match Legitimate Name or Location )
 start %temp%\audiodg.exe
 dir d:\ >> %temp%\download
 systeminfo >> %temp%\download
 del %0
```

So here we can see (ignoring the sequence) a list of techniques that represent the whole attack.

We can compare this list with the group technique sets in ATT&CK and try and discover a likely attacker. An attempt at this approach looking for the APT with the maximum number of matching techniques is described in the following paragraphs (see also Validation Of Simple Exact Matching).

The extract below shows a simple result from a query on the ATT&CK knowledge base. As mentioned above it simply looks for the APT with the maximum number of techniques matched.

The value of this result seems low but is included as an illustration of the direction of development. In this illustrative example, knowledge about the group is derived from only one attack report and all techniques identified in this report (see extract above). In this example the set of techniques matches exactly the ATT&CK technique set provided for the APT admin@338.

A utility was created and used (see (Maidens, 2023)) to return all groups who use techniques that match those in a given example attack sequence. These are sorted in the order of number of matches found. Here the example attack sequence is shown at the start of the output with APT matches shown below.

```
<< STARTING ATTACK DB TEST APPLICATION
Technique - T1566.001 Spearphishing Attachment is part of Tactic - TA0001 Initial Access
Technique - T1204.002 Malicious File is part of Tactic - TA0002 Execution
Technique - T1059.003 Windows Command Shell is part of Tactic - TA0002 Execution
Technique - T1083 File and Directory Discovery is part of Tactic - TA0007 Discovery
Technique - T1082 System Information Discovery is part of Tactic - TA0007 Discovery
Technique - T1016 System Network Configuration Discovery is part of Tactic - TA0007 Discovery
Technique - T1007 System Service Discovery is part of Tactic - TA0007 Discovery
Technique - T1069.001 Local Groups is part of Tactic - TA0007 Discovery
Technique - T1049 System Network Connections Discovery is part of Tactic - TA0007 Discovery
<<<< ------------------------
```

```
<<<< Best matched results Test01 >>>>
<<<< Expected result admin@338 >>>>
<<<< ------------------------
{'GroupName': 'admin@338', 'TTPCount': 9, 'TTPList': ['T1566.001', 'T1204.002', 'T1059.003',
'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049']}
<<<< - All results are noted below
<<<< - Total number of matches is 99
<<<< - Total number of possible groups is 134
<<<< - Set 1
Name is : admin@338
Match count is : 9
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001',
'T1049']
```

6.3.2.1 Validation Of Simple Exact Matching

The above simple example works in this case because the list of techniques in the 'attack fragment' matches the list of techniques recorded for this APT (in MITRE ATT&CK).

But it is easy to see how this approach only offers limited additional intelligence.

In fact, a total of 99 possible APT technique sets have a partial matching.

For instance, the next best 'match' is with eight techniques (OilRig (Set 2)).

Additionally, there are a number of APT with technique sets matching seven of the techniques present in this attack segment. Within this we can see APT32 (Set 3), Lazarus Group (Set 5), MuddyWater (Set 6), Mustang Panda (Set 7), Sandworm Team (Set 8) and Tropic Trooper (Set 9) are matching on the exact same set of techniques.

The relevant output is shown below :-

```
<<<< - Set 2
Name is : OilRig
Match count is : 8
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049']
<<<< - Set 3
Name is : APT32
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1049']
<<<< - Set 4
Name is : Kimsuky
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1007']
<<<< - Set 5
Name is : Lazarus Group
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1049']
<<<< - Set 6
Name is : MuddyWater
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1049']
<<<< - Set 7
```

```
Name is : Mustang Panda
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1049']
<<<< - Set 8
Name is : Sandworm Team
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1049']
<<<< - Set 9
Name is : Tropic Trooper
Match count is : 7
TTP list is : ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1049']
<<<< - Set 10
Name is : Chimera
Match count is : 7
TTP list is : ['T1059.003', 'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049']
<<<< - Set 11
Name is : Operation Wocao
Match count is : 7
TTP list is : ['T1059.003', 'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049']
<<<< - Set 12
Name is : Turla
Match count is : 7
TTP list is : ['T1059.003', 'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049']
```

Some obvious limitations become apparent quite quickly.

Firstly, we may have only ‘sensed’ a few of the techniques that are actually being used in the attack.

> The integration of ATT&CK and CAR to sense and deliver an observed technique stream is described above (Cyber Analytics Repository (CAR)) .
>
> The smaller the set of techniques sensed the more likely we are to match multiple APTs (or even miss likely candidates) limiting specific awareness that can be achieved.

Secondly, we may only have a partial coverage of the techniques used by APTs in the intelligence we have to hand from ATT&CK.

> Manual checking of the reports within ATT&CK do show examples of technique usage that have not been recorded within the knowledge base (presumably resulting from the inevitable limit to time available to manually update ATT&CK as new reports are published).

Thirdly, (and most relevant here) the APT technique sets are a synthesis of multiple reports, and this obscures the specific sequencing of techniques within the individual attacks.

Fourthly, at this point we are only matching on APT technique sets only.

> In fact, some of the techniques used by APTs in an attack are implemented via tools. Despite being executed through tooling these should still be considered as part of the attacker's technique decision set.

6.3.3 Conclusions From Simple Example

Simply keeping a list of techniques used by each APT is useful but this removes the intelligence available from openly available intelligence (e.g published attack reports) describing the sequence of techniques used in particular cyber-attacks.

MITRE themselves discourage users from trying to use the ATT&CK dataset to 'attribute' an attack to a particular APT. So, in the example shown above where we match the techniques in the attack segment to a particular APT the intelligence gained is very generalised.

The way the ATT&CK Tactics and Techniques are structured does not link directly to a sequential kill chain view. Rather they represent 'blocks' of related behaviours that can be relevant to multiple phases of an attack. A number of Techniques are relevant to multiple Tactics. Tactics such as 'Command and Control' and 'Execution' may be used across multiple phases of a sequential Kill Chain model.

In the six APT technique sets described above that all match seven of the techniques present in the example attack segment there maybe differences in the sequencing used in the attacks that would allow us to make more refined decisions about matches to previous attacks that we can use to reason possible appropriate mitigation actions, possible additional impacted system areas and possible next steps that the attacker may try to make.

6.4 An Initial Attempt at Recording Attacks as Sequences

6.4.1 Introduction

This is the next step taken to define an initial attack sequence model as outlined in Introduction.

The work above (Motivating Example) motivates investigation into whether additional intelligence can be gleaned by recording known APT attacks as sequences of ATT&CK tactics and techniques. An initial attempt at an attack sequence model is described here.

6.4.2 Initial Attempt at a Model

6.4.2.1 Introduction

For each of the attack reports available for the APTs we want to create a model that can contain a description of the attack(s) as a stream of sequenced tactics and techniques (one for each attack). This section provides a summary of an initial version of this model.

The model described in this section draws on some of the general approaches outlined in (Takahashi et al., 2020) and (Choi et al., 2021) who consider the automatic generation of attack sequences to support incident response prioritisation. The model (developed in this chapter) provides a format to allow the description of each step in a sequence (of tactics and techniques) used by an attacker. Each step describes the technique (and associated tactic) used by the attacker. Software use is not maintained in this model as the emphasis of this investigation is on the complete set of actions taken by the attacker. Any software used by the attackers is simply seen by the tactics and techniques observed through its use. The steps taken together are used to describe an attack.

Each step in an attack will be described using the following fields:

- **ID** – Used to identify the step sequence number.
- **Tactic** – The tactic identified for the relevant technique (this may be NULL).
- **Technique** – The technique identified for this step.
- **Pred** – Predecessor step.
- **KC Step** – At this time this will be populated with appropriate elements of the Unified Kill Chain.
- **Notes** – Reference to source material/justification.

These are presented as ‘rows’ in the ‘table’ and are given in the example below.

6.4.2.2 Example for admin@338

For the admin@338 attack example given above (see A Motivating Simple Initial Example), and manual analysis gave the following sequence of tactics and techniques to describe the attack (an appropriate kill chain will be addressed later).

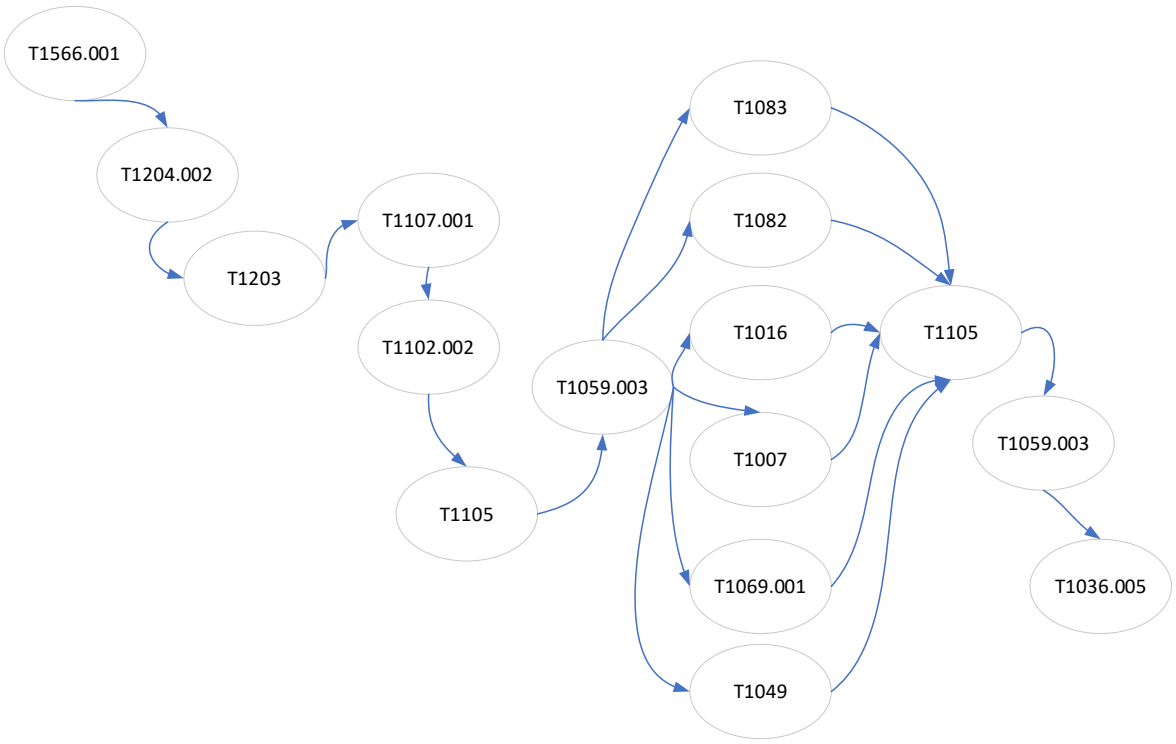
	Chain_ID	***admin@338_001***			
ID	***Tactic***	***Technique***	***Pred***	***Kill Chain Step***	***Notes***

1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**			
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		Includes malicious file (backdoor Lowball)
3	**TA0002 : Execution**	**T1203 : Exploitation for Client Execution**	2		The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code)
4	**TA0011 : Command & Control**	**T1107.001 Application Layer Protocol : Web Protocols**	3		Detail related to above
5	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	4		Initial installation connects to C&C
6	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	5		Install upgraded tool
7	**TA0002 : Execution**	**T1059.003 Command and Scripting Interpreter: Windows Command Shell**	6		Still part of Initial Access step. Commands executed to achieve initial Discovery (etc aims)
8	**TA007 Discovery**	**T1083 File and Directory Discovery**	7		
9	**TA007 Discovery**	**T1082 System Information Discovery**	7		
10	**TA007 Discovery**	**T1016 System Network Configuration Discovery**	7		
11	**TA007 Discovery**	**T1007 System Service Discovery**	7		
12	**TA007 Discovery**	**T1069.001 Permission Groups Discovery: Local Groups**	7		
13	**TA007 Discovery**	**T1049 System Network Connections Discovery**	7		
14	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	8-13		Install second stage tool (Bubblewrap)
15	**TA0002 : Execution**	**T1059.003 Command and Scripting Interpreter: Windows Command Shell**	14		To install second stage tool above
16	**TA005 Defense Evasion**	**T1036.005 Masquerading: Match Legitimate Name or Location**	15		Rename second stage tool with benign name

We can also simplify this further and observe the attack as a Tactic chain, as shown below

Tactic
TA0001 : Initial Access
TA0002 : Execution
TA0011 : Command & Control
TA0002 : Execution
TA007 Discovery
TA0011 : Command & Control
TA0002 : Execution
TA005 Defense Evasion

We also note that the technique chain above can be seen in graph-like form below. The 'nodes' represent each of the steps in the table above. There is no intention to imply cyclic structures within this representation although future work could consider this possibility.



This in turn can be codified to be used with in Python (for example) as shown below

[{"ID":"admin@338_001"},
{"StepNo":"1", "Tech":"T1566.001", "Pred":"0"},
{"StepNo":"2","Tech":"T1204.002", "Pred":"1"},
{"StepNo":"3","Tech":"T1203", "Pred":"2"},
{"StepNo":"4","Tech":"T1102.002", "Pred":"3"},
{"StepNo":"5","Tech":"T1071.001", "Pred":"4"},
{"StepNo":"6","Tech":"T1105", "Pred":"5"},
{"StepNo":"7","Tech":"T1059.003", "Pred":"6"},
{"StepNo":"8","Tech":"T1083", "Pred":"7"},
{"StepNo":"9","Tech":"T1082", "Pred":"7"},
{"StepNo":"10","Tech":"T1016", "Pred":"7"},
{"StepNo":"11","Tech":"T1007", "Pred":"7"},
{"StepNo":"12","Tech":"T1069.001", "Pred":"7"},
{"StepNo":"13","Tech":"T1049", "Pred":"7"},
{"StepNo":"14","Tech":"T1105", "Pred":"8-13"},
{"StepNo":"15","Tech":"T1059.003", "Pred":"14"},
{"StepNo":"16","Tech":"T1036.005", "Pred":"15"}]

For this work we are building knowledge of the attack from techniques detected, so we may have no knowledge of software being used by the attacker.

6.4.3 Review of the Initial Attempt

This structure was used with above example and a few additional reports (examples from the APT groups Lazarus Group and APT32) and some observations were made. These are outlined below and resulted in changes to the model (that are described in A New Attack Model).

In the 'Discovery' (TA007) section of the attack the original decision had been to simply describe the related Techniques as a group (Step 8 to 13) and recording the same previous step. The following step (Step 14) then logically followed this group. This was not carried forward on the assumption that these Techniques will actually be performed in some kind of sequence either manually or via automation. This provides more detail on the precise sequence within the attack and simplifies the model for the attack description.

Although timings are not generally available in open source reporting a timings column has been added for future proofing. This would provide increased insight into the temporal qualities of the various attacks. The rationale for this is described in a little more detail below in the section discussing general approaches to 'measuring' similarity between sequences.

Several techniques observed (especially within the Command & Control phase descriptions) do not actually represent sequential elements of the attack. For instance we may have intelligence on the web protocols being used (e.g. T1071.001 Application Layer Protocol : Web Protocols) but this is supporting information about the action of the sending/receiving a message. For this reason, an additional label was added (S/G) to indicate if this is a sequential step (S) or not. The

'non-S steps' preceding this are then used as additional intelligence that can be used for comparison purposes.

A meta model is also required to provide more intelligence to classify attacks when in a database of other similarly recorded attacks. This will allow future high-level analysis of the types of attack intelligence collected.

6.5 A New Attack Model

6.5.1 Introduction

This is the next step taken to define an attack sequence model as outlined in Introduction.

Based on the observations (see Review of the Initial Attempt) from the initial attempt at the model (see An Initial Attempt at Recording Attacks as Sequences), the overall model was refined further.

- A new Attack Sequence Model (see The Refined Attack Sequence Model is now described.
- A Meta Model (see The Meta Model) is also added and described. This is used to index the recorded attacks including documentation of some of the qualities of the attack that cannot be understood simply through observation of the tactics and techniques.
    - As this Meta Model is indexing the attack sequences, a refined definition of a cyber-attack (assumed in both the meta model and attack sequence model) is provided in this section. This is summarised in the bullets below.
    - In (Derbyshire et al., 2018) we have "**we can consider a cyber-attack to be an offensive action taken against a target's cyber infrastructure. This includes connected computers, software, networks, procedures, and people**". This definition is made a bit more precise here to allow smaller units of attack to be recorded and linked. Within the Derbyshire description a smaller unit of the attack can be defined and then these smaller units (the attack sequences) can be linked to provide a description of the whole attack.
    - **Each of the sequences referenced by the Meta Model here, will describe the sequence of actions from initial access to a targeted asset or to the point pivot/lateral movement is achieved (if this occurs).**

6.5.2 The Meta Model

A meta model was introduced as follows (the rationale for this is given below). Additional details are provided in Comparing the Initial and New Attack Model.

As mentioned in the Introduction (see Introduction) above, each of the sequences here will describe the sequence of actions from initial access to a targeted asset or to the point pivot/lateral movement is achieved (if this occurs).

This provides a few advantages:

- Firstly, the scope of the sequence documented is understood.
    - Attack reports often include descriptions of attacks across a whole target. Where lateral movement is achieved this may mean that the attacker may achieve multiple parallel 'effects' on the target.
- Secondly, it means that these sequences can be kept as a sequential series of actions.
- Thirdly, by linking sequences (this is achieved through the Preceded By field) support is given for multi-step attacks (see also The Meta Data Prec Field).
    - An attack sequence begins with either initial access or following pivot/lateral movement. It ends at access to targeted asset of pivot/lateral movement.
        - "**Pivoting** describes the act of tunnelling traffic through one system to connect to other internal systems" (Pols, 2017)
        - "In the **lateral movement** phase of APT campaigns the attackers try to compromise additional systems within the internal environment of the target" (Ussath et al., 2016)
    - An attack made of multiple steps can be described by linking multiple sequences describing attack sequences between initial access and pivot/lateral movement. This includes blended attacks – which is where an attacker may deploy multiple approaches and exploits following initial access.

The general Meta-Model description of a cyber-attack sequence is given here

Dimension	***Description***	***Notes***
Attribution	Possible APT	From Controlled List This will also include generic descriptions of type of attacker (e.g. APT, Cyber Criminal etc)

Initial Access Vector	The MITRE Initial Access vector	List ATT&CK Initial Access Vectors. A more precise characterisation of the attack can be derived from the associated technique sequence.
Attack Origin	Potential 'country' source of attack. These cannot be considered as true attribution but will be based on ATT&CK/TCert intelligence.	From Controlled List
Target Location	The location of the actual attack. The location of organisation is intended here.	From Controlled List
Target Type	Type of organisation	From List
Impact	One of: • Exfiltration (confidentiality) • Damage (integrity and availability) • Reputation (external to the target)	From Controlled List
Vulnerabilities Exploited	CVE-????-????, CWE-????	List CVE/CWE
Related Attack Patterns	Provides ability to connect campaign elements.	List
Preceded By	Provides support for multi-step attack descriptions by linking stages of attacks. It also allows for parallel elements of an overall attack.	Reference Sequence_ID

Schema Version	0.1	To allow for future new sequence models
Date	Date of attack. To give some sense of currency on information. A granularity of year will be used.	Matching attack fragments may favour recent attacks over older attacks

An example populating the general model given above with real data. is given here.

Dimension	***Technique***	***Notes***
Attribution	admin@338	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	China	
Target Location	Hong Kong	
Target Type	Media	
Impact	Exfiltration	Monitoring media orgs
Vulnerabilities Exploited	CVE-2012-0158	
Related Attack Patterns	TBC	
Preceded By	TBC	
Schema Version	0.1	
Date	2015	

6.5.3 The Meta Model Rationale

6.5.3.1 Purpose

The purpose of the meta model is to both provide an indexing for the attacks in an eventual knowledge base of attack sequences but also to provide additional high-level intelligence that can be used to observe trends.

6.5.3.2 Dimensions

This section describes how this proposed Meta Model is built on previous research proposals and extended to this specific application.

Based on the work in (Hansman & Hunt, 2005) a dimension based approach is used.

Of the original dimensions the following were used as input.

- Attack Vector
    - Renamed Initial Access Vector
    - This is represented by the ATT&CK initial access technique used within the attack.
    - In some attacks it is clear that initial access is achieved using a combination of Initial Access techniques (e.g. APT28_0001) so support for a list is relevant here
- Classification of Target
    - Renamed Target Type
    - The original intention in (Hansman & Hunt, 2005) was defined as the technical component (or class of component) being attacked.
    - In this model this is replaced by classification of the organisation being attacked.
    - Adding additional context such as Asset Types is suggested in the Future Work section. However, this may be better associated with steps in the subsequent attack sequence description. This more accurately describes the attack 'trajectory'. Asset details associated with the Initial Access step would naturally substitute for the original meaning.
- Vulnerabilities Attacked
    - Renamed Vulnerabilities being exploited
    - Vulnerabilities and exploits being used in this attack.
    - Adding additional context such as Vulnerabilities exploited at step level is suggested in the Future Work section. This more accurately describes the attack 'trajectory'.

'Payloads' (originally included in (Hansman & Hunt, 2005)) is not included as the focus here is on the technique sequences. Adding additional context such as Tool/Software is suggested in the Future Work section (see Future Work).

Based on (Simmons et al., 2009) the following were used as input

Defence was removed as this is covered via links to ATT&CK Technique Mitigations and MITRE D3FEND.

- Operational and Information Impact
    - Renamed as Impact and simplified to one of Exfiltration, Damage or Reputation

In (Pöhn & Hommel, 2022) the Attacker Types and Capabilities are covered by the ATT&CK intelligence being used. All types here are classed as APT and capabilities are reflected in the Technique sequences.

- Attribution
    - Currently this is just the ATT&CK source 'attribution'. Future use could include a broader set of Actors or even a general suggested attacker type (based on available intelligence) e.g., Cyber Criminal.

From (S. Kim et al., 2019) the Attack Procedures is covered by the Technique sequences (described below)

Over and above these elements some additional meta data elements are added.

- Attack Origin
    - Where reliable or available, this extends Attribution and provides input to trend analysis.
- Target Location
    - Extending Target Type and providing input to trend analysis
- Related Attack Patterns
    - To allow building of Campaign intelligence
- Preceded By
    - To allow chaining for multi-step attacks and possible parallel attack sequences.
- Date
    - Providing input to trend analysis, also potential input to likelihood in pattern matching
- Schema Version
    - Allows future development of Technique Sequence schema

6.5.3.3 Sense Checking the Meta Model

From (Hansman & Hunt, 2005) we have a set of criteria that we can use to 'sense check' a taxonomy.

A brief discussion of how these criteria relate to the Meta Model is provided here and have been used to provide a sense check of the model.

- **Accepted**: The taxonomy should be structured so that it can become generally approved.
  - This is structured in this way
- **Comprehensible**: A comprehensible taxonomy will be able to be understood by those who are in the security field, as well as those who only have an interest in it.
  - The rationale for the dimensions is described above and are constructed from defined terms
- **Completeness**: For a taxonomy to be complete/exhaustive, it should account for all possible attacks and provide categories accordingly. While it is hard to prove a taxonomy that is complete or exhaustive, it can be justified through the successful categorisation of actual attacks.
  - This has been tested against the representative attacks documented within ATT&CK
- **Determinism**: The procedure of classifying must be clearly defined.
  - The rationale for the dimensions is described above
- **Mutually exclusive**: A mutually exclusive taxonomy will categorise each attack into, at most, one category.
  - This is not relevant to this application
  - It may be useful to separate the classification and meta data elements
- **Repeatable**: Classifications should be repeatable.
  - The same intelligence will create the same classification
- **Terminology complying with established security terminology**: Existing terminology should be used in the taxonomy to avoid confusion and to build on previous knowledge.
  - Care has been taken to define terms in preparation for this model
- **Terms well defined**: There should be no confusion as to what a term means.
  - Care has been taken to define terms in preparation for this model
- **Unambiguous**: Each category of the taxonomy must be clearly defined so that there is no ambiguity with respect to an attack's classification.
  - The rationale for the dimensions is described above
- **Useful**: A useful taxonomy will be able to be used in the security industry and particularly by incident response teams.
  - In the Future Work section (see Future Work) development of an Open-Source Knowledge Base is described. This would provide an opportunity to address this point.

6.5.4 The Refined Attack Sequence Model

6.5.4.1 The Refined Attack Sequence Description

Based on review of (see Review of the Initial Attempt) the Initial Attack Sequence (see Initial Attempt at a Model) model was refined.

The original model fields were as follows:

- **ID** – Used to identify the step sequence number.
- **Tactic** – The tactic identified for the relevant technique (this may be NULL).
- **Technique** – The technique identified for this step.
- **Pred** – Predecessor.
- **KC Step** – At this time this will be populated with appropriate elements of the Unified Kill Chain.
- **Notes** – Reference to source material/justification.

Two new fields have been added (see also Recording Attacks as Sequences):

- **Tinc** – This is not used at this point but was added to provide for future refinements (see Future Work).
    - It is intended to allow timings of each functional step (offset from the time of initial access) to be recorded.
    - This was motivated by the recognition that a sequence of the same techniques in two separate attacks may be actioned with different timings. These could be used to detect separate behavioural characteristics.
- **S/G** – This field was added to distinguish between techniques that described functional steps taken by attackers (S) and techniques that described general qualities of the functional step (G).

A change in approach has been made to the use of one field (see also Recording Attacks as Sequences):

- **Pred** – Previously for contiguous steps recording Discovery type techniques this was being treated as a parallel set up steps by pointing to the 'Predecessor' execution technique being used to run them (see Example for admin@338). This approach is no longer used. Instead, the sequence that they are run in is recorded (see the amended example below).

Additional detail for these changes is also provided in Comparing the Initial and New Attack Model.

An example of the refined attack model is shown below.

The colouring is used as an indicator to show which techniques are present in the ATT&CK APT descriptions. The full set of attacks is shown in the Appendix below with additional explanation. The Future Work section also describes possible additional fields to be considered at a later point

Here we have

- **ID** – Used to identify the step sequence number.
- **Tactic** – The tactic identified for the relevant technique (this may be NULL).
- **Technique** – The technique identified for this step.
- **Pred** – Predecessor (used to allow information 'G' steps to be included).
- **Tinc** – Time Increment (not used currently but intended to support future analysis of potential timing patterns in attacks).
- **S/G** – An action taken is represented as a step (S). General information associated with a step is represented with a G.
- **KC Step** – At this time this will be populated with appropriate elements of the Unified Kill Chain
- **Notes** – Reference to source material/justification.

	Sequence_ID	***admin@338_001***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**			S		
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S		Includes malicious file (trojan downloader Lowball [2])
3	**TA0002 : Execution**	**T1203 : Exploitation for Client Execution**	2		S		The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code)

4	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	3				This technique to provides more detail on step below
5	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	3		S		Initial installation connects to C&C
6	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	5		S		Install upgraded tool
7	**TA0002 : Execution**	**T1059.003 Command and Scripting Interpreter: Windows Command Shell**	6		S		Still part of Initial Access step. Commands executed via .bat. Discovery
8	**TA007 Discovery**	**T1083 File and Directory Discovery**	7		S		
9	**TA007 Discovery**	**T1082 System Information Discovery**	8		S		
10	**TA007 Discovery**	**T1016 System Network Configuration Discovery**	9		S		
11	**TA007 Discovery**	**T1007 System Service Discovery**	10		S		
12	**TA007 Discovery**	**T1069.001 Permission Groups Discovery: Local Groups**	11		S		
13	**TA007 Discovery**	**T1049 System Network Connections Discovery**	12		S		

14	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	13		G		This technique to provide more detail on step below
15	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	13		S		Initial installation connects to C&C
16	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	15		S		Install second stage tool (Bubblewrap)
17	**TA0002 : Execution**	**T1059.003 Command and Scripting Interpreter: Windows Command Shell**	16		S		To install second stage tool above
		Techniques unclear here					We know [2] The BUBBLEWRAP malware is installed with admin rights and the threat actors gain full access to the compromised machine [2] The BUBBLEWRAP Trojan may create a hidden system administrator account.
	TA005 Defense Evasion	**T1036.005 Masquerading: Match Legitimate Name or Location**	15		G		Rename second stage tool with benign name. But this is on the DropBox server

18	**TA007 Discovery**	**T1082 System Information Discovery**	17		S		
19	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	18		S		Bubblewrap communications (not via Dropbox server, this Tech noted in ATT&CK)

Table 7 - admin@338_001 Attack Sequence Example

Some code was also developed to semi-automate conversion of the Word tables into a python readable form (see (Maidens, 2023)). As shown here

[
{'ID':'admin@338_001', 'Version':'0.1'},
{'StepNo':'1','Tactic':'TA0001','Tech':'T1566.001','Pred':'0','Tinc':'','SG':'S','KC':''},
{'StepNo':'2','Tactic':'TA0002','Tech':'T1204.002','Pred':'1','Tinc':'','SG':'S','KC':''},
{'StepNo':'3','Tactic':'TA0002','Tech':'T1203','Pred':'2','Tinc':'','SG':'S','KC':''},
{'StepNo':'4','Tactic':'TA0011','Tech':'T1071.001','Pred':'3','Tinc':'','SG':'G','KC':''},
{'StepNo':'5','Tactic':'TA0011','Tech':'T1102.002','Pred':'3','Tinc':'','SG':'S','KC':''},
{'StepNo':'6','Tactic':'TA0011','Tech':'T1105','Pred':'5','Tinc':'','SG':'S','KC':''},
{'StepNo':'7','Tactic':'TA0002','Tech':'T1059.003','Pred':'6','Tinc':'','SG':'S','KC':''},
{'StepNo':'8','Tactic':'TA0007','Tech':'T1083','Pred':'7','Tinc':'','SG':'S','KC':''},
{'StepNo':'9','Tactic':'TA0007','Tech':'T1082','Pred':'8','Tinc':'','SG':'S','KC':''},
{'StepNo':'10','Tactic':'TA0007','Tech':'T1016','Pred':'9','Tinc':'','SG':'S','KC':''},
{'StepNo':'11','Tactic':'TA0007','Tech':'T1007','Pred':'10','Tinc':'','SG':'S','KC':''},
{'StepNo':'12','Tactic':'TA0007','Tech':'T1069.001','Pred':'11','Tinc':'','SG':'S','KC':''},
{'StepNo':'13','Tactic':'TA0007','Tech':'T1049','Pred':'12','Tinc':'','SG':'S','KC':''},
{'StepNo':'14','Tactic':'TA0011','Tech':'T1071.001','Pred':'13','Tinc':'','SG':'G','KC':''},
{'StepNo':'15','Tactic':'TA0011','Tech':'T1102.002','Pred':'13','Tinc':'','SG':'S','KC':''},
{'StepNo':'16','Tactic':'TA0011','Tech':'T1105','Pred':'15','Tinc':'','SG':'S','KC':''},
{'StepNo':'17','Tactic':'TA0002','Tech':'T1059.003','Pred':'16','Tinc':'','SG':'S','KC':''},
{'StepNo':'18','Tactic':'TA0007','Tech':'T1082','Pred':'17','Tinc':'','SG':'S','KC':''},
{'StepNo':'19','Tactic':'TA0011','Tech':'T1102.002','Pred':'18','Tinc':'','SG':'S','KC':''}

]

6.5.4.2 Attacks as Sequences of Events

This work makes the broad assumption that attacks can be usefully described as a sequence of events. In this case the events are described as ATT&CK techniques. This section will seek to qualify this assumption.

This approach differs from broader formal attack languages. It is intended to provide a model of the techniques used in a specific attack. It does not intend to model all possible attack techniques through an attack graph. Nor does it intend to model attack campaigns where an attacker may take multiple (AND/OR) technique decisions at various points.

The reason for keeping it in this form is to allow analysts (in the future) to record sequences alongside pure textual descriptions of attacks. ATT&CK Techniques are increasingly being embraced as a standard 'language' to describe attacks. Taken together these can build into a substantial knowledge base of attacks that would provide a valuable source of intelligence to analysts and researchers.

This can be seen as a mechanism to help bridge between a high-level sequenced Kill Chain based view of an attack and the mid-level view offered by MITRE ATT&CK and the non-sequenced Tactics model.

Fundamental to the design of the many Kill Chain models (outlined above) is the consideration that the attacker's actions are following a sequence of high-level steps. This is not without criticisms but provides some framework for planning mitigations in response to detected attacks. This concept is discussed a little further in (Howard & Olson, 2020) and the creation of playbooks. Attack sequences describing attacks are also discussed in (Eckmann et al., 2002).

As an attacker proceeds through the various phases of an attack kill chain lifecycle, they will make use of Tactics and Techniques in sequence to achieve their sub-goals. It is possible that an attacker may build multiple pivots and achieve lateral movement however each of these 'units of work' can be seen simplistically as a sequence of actions at the Technique and Tactic level (see also Future Work section). Certainly, these sequences can be seen as valuable signatures that can provide an additional source of actionable intelligence.

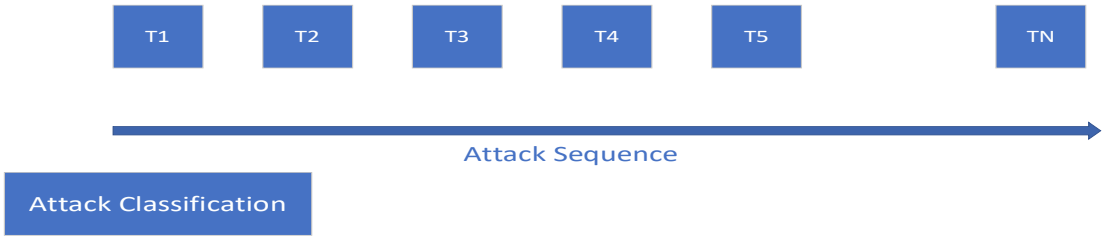
The abstract attack modelling approaches that have been variously proposed (see Attack Modelling Languages) generally seek to model all possible attack ***paths*** that **may potentially** be used by an attacker so that defences/mitigations can be organised. The attack sequences to be modelled here differ as they are required to record the **actual *sequences*** used by attackers in

specific attacks (some support for Campaigns is provided by allowing attack sequences to be grouped via the 'Related Attack Patterns' field in the 'Attack Categorisation' metadata).

6.5.4.2.1 Recording Attacks as Sequences

For this work we are assuming a future situation (with justification provided in the Related Work section) that there exists a system that can detect 'noisy' sequences of techniques attributable to an attack.

Known attacks are recorded in the following standardised format. To this author's knowledge this format with this broad use case has not been proposed in research papers elsewhere.



As described above each element TN in the Attack Sequence is recorded as a row in a table as follows:

ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*

Where

ID	Is a unique id for this row/step in the sequence
Tactic	MITRE Tactic associated with this Technique. MITRE Techniques do not have a one-to-one mapping with MITRE Tactics.
Technique	MITRE Tactic associated with this step
Pred	Predecessor 'functional' step (S see below)
Tinc	Not currently used, timestamp increment in this attack. Intended for future use as timings between steps may provide additional intelligence beyond the simple sequencing itself (see Related Work)
S/G	Whether this step is a real functional step or supporting information. Examples here include the C&C and Defense Evasion techniques that provide knowledge about qualities of an action being performed (e.g. encrypted messaging or

	obfuscated code) but not the action itself (e.g. send message or deobfuscation of code prior to execution)
KC Step	(In these examples) UKC phase associated with this step. In some case multiple phases are relevant such as with Spearphishing techniques where both Delivery and Social Engineering phases are implicit. These can be provided in a list (/ separated in the testing code)
Notes	Supporting evidence where relevant. These are used extensively here as a large amount of manual work has been carried out to create the test set being used here

As will be described in more detail later (Using the Attack Model – Hidden Markov Model) the Tactic (and possibly KC Step) may be considered as potentially hidden states when only Techniques can be detected.

6.6 Loading a Representative Data Set

6.6.1 Introduction

This is the next step taken to define an attack sequence model as outlined in Introduction.

Following the initial testing above, this section seeks to further demonstrate justification for the model proposed by loading a representative set of attacks.

The 8 example attacks documented and loaded so far are given in the table below (see also Appendix E)

1	*admin@338_001*
2	*Lazarus_Group_001*
3	*Lazarus_Group_002*
4	*APT32_001*
5	*MuddyWater_001*
6	*MuddyWater_002*
7	*Mustang_Panda_001*
8	*Sandworm_001*

9	*Tropic_Trooper_001*

Table 8 - Interim Test Attacks

In this section a further 17 attack sequences are added with a demonstration of the coverage obtained.

8 are added in An Analysis Against ATT&CK Tactics/Techniques and Adding Further Attacks to Complete Tactic and UKC Phase Coverage and a further 9 in Adding Further Attacks to Demonstrate Additional APT Groups

6.6.2 Loading the New Attack Model into an Example Database

An example approach given here considers loading the attack sequences into a graph database. The model outlined below was used to persist (using the python package NetworkX) the attack sequences. This was used in some of the examples shown in Results. Although it is just one example of how the data may be persisted, the edges within the graph model provide quick access to statistics on relationships between tactics/techniques through analysis of the relevant edge counts. These statistics can provide information about probabilities of likely transitions between Techniques observed in attacks (e.g. Using the Attack Model – Hidden Markov Model). A toy example to illustrate this is given in Small Note on Implementation of Graph Model.

MultiDiGraph—Directed graphs with self-loops and parallel edges

The nodes and edge 'schema' is given here (in GEXF format)

```
<graph defaultedgetype="directed" mode="static" name="">
  <attributes mode="static" class="edge">
    <attribute id="3" title="attack_id" type="string" />
    <attribute id="4" title="count" type="long" /> # Equals attack step
    <attribute id="5" title="name" type="string" />
    <attribute id="6" title="create_date" type="string" />
    <attribute id="7" title="edgetype" type="long" /> # Chain type - 0 Tech, 1 Tech Supp, 2 Tactic,
                                                      3 Tech Tactic, 4 Supp Tech Tactic
    <attribute id="8" title="networkx_key" type="long" />
  </attributes>
  <attributes mode="static" class="node">
    <attribute id="0" title="name" type="string" />
    <attribute id="1" title="create_date" type="string" />
    <attribute id="2" title="type" type="long" /> # 1 Metadata, 2 Tech, 3 Tactic

  </attributes>
```

Node Types

1 - Attack Instance – Equates to Sequence ID

2 - Tactic – Equates to MITRE Tactic

3 - Technique – Equates to MITRE Technique

Edges

Each edge identifies attack name and step ID

Type 0 – Tech Sequence Edge (Attack Instance->Technique, Technique->Technique)

Technique steps of attack

Type 1 – Tech Support Edge (Technique->Technique)

Techniques providing additional intelligence to Step Technique

Type 2 – Tactic Sequence Edge (Attack Instance->Tactic, Technique->Tactic)

Tactic steps of attack

Type 3 – Technique Tactic Edge (Technique->Tactic)

Tactic associated with Technique in an attack

Type 4 – Support Technique Tactic (Technique->Tactic)

Tactic associated with Support Technique in an attack

A simplified view of the start of the above example (see The Refined Attack Sequence Description) is illustrated here

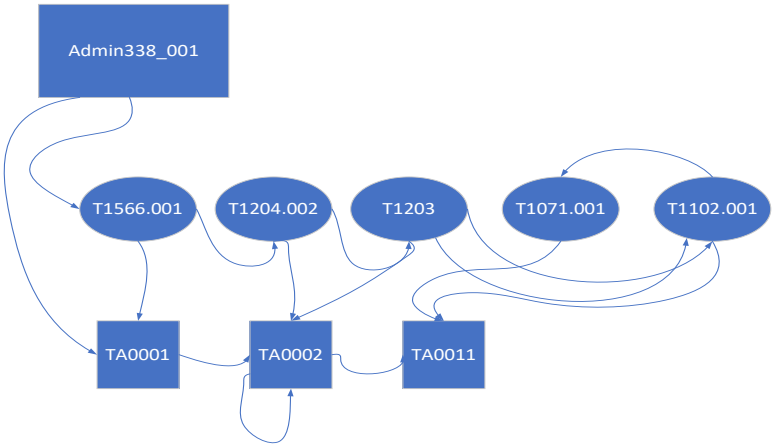


Figure 81 - Attack Sequence Graph Example

6.6.3 An Analysis Against Unified Kill Chain

This section demonstrates how various Kill Chain structures can be supported using the proposed sequence model.

This will be demonstrated against the high level structure presented in the Unified Kill Chain (UKC) (Pols, 2017) (also explained in the Background section above) and a number of Case Studies. The outline structure of the UKC is show below

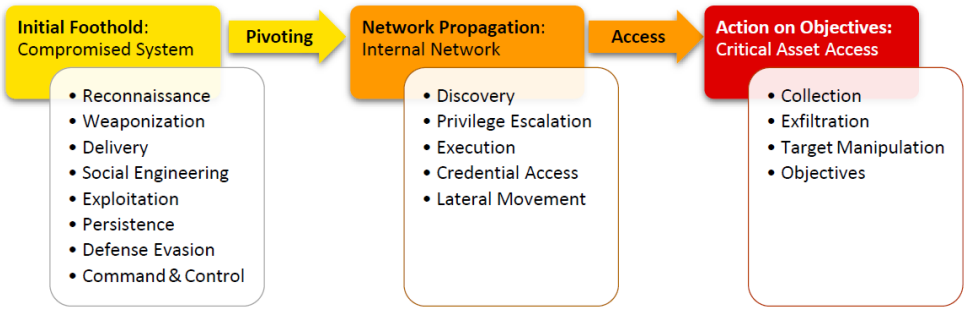


Figure 82 - UKC Summary

This model was chosen as it develops and unites the original Lockheed Martic CKC presented in (Hutchins et al., 2011) with refinements presented by Laliberte (Laliberte, 2016), Nachreiner (Nachreiner, 2015), Bryant (Bryant & Saiedian, 2017), Malone (Malone, 2016) and considerations raised by MITRE ATT&CK (Strom et al., 2020).

The sequential model presented here is intended to provide a common approach to recording cyber-attacks detected by organisations. For this reason, the Reconnaissance and Weaponization elements have been excluded from scope. The motive for removing Weaponization actually

accords with a number of analyses e.g. Nachreiner (Nachreiner, 2015) where it is argued that no action can be taken by 'defender' to mitigate this. The motive for removing Reconnaissance is the limited sequential information provided by this phase (although some forms of reconnaissance (e.g. simple pings) may be detectable).

The Objectives element within Action on Objectives was added to UKC to provide for socio-technical objectives of an attacker. This will also be left out of scope for this work.

The precise distinction between Pivot and Lateral Movement in the UKC should be noted. This distinction is noted in Nachreiner (Nachreiner, 2015). From (Pols, 2017) we have "Pivoting may be regarded as an implicit technique that is enabled by (and part of) Command & Control and is required to perform further actions such as Discovery and Lateral Movement". The distinction here is between the attacker using a platform such as a Backdoor to execute techniques and the actual movement and development of access to additional target platforms. This distinction is not always so precise in other works and often the two terms are used synonymously.

Short-hand references will be used for the various UKC elements and are enumerated below

Initial Foothold

IF-REC, IF-WEP, IF-DEL, IF-SEN, IF-EXP, IF-PER, IF-DEV, IF-C2C

Network Propagation

NP-DIS, NP-PES, NP-EXE, NP-CAC, NP-LMV

Action on Objectives

AO-COL, AO-EXF, AO-TMA, AO-OBJ

We can understand the 'coverage' obtained from the attack sequences used to date

	IF-REC	IF-WEP	IF-DEL	IF-SEN	IF-EXP	IF-PER	IF-DEV	IF-C2C	NP-DIS	NP-PES	NP-EXE	NP-CAC	NP-LMV	AO-COL	AO-EXF	AO-TMA	AO-OBJ
1			x	x	x	x		x	x		x						
2			x	x	x		x	x	x		x				x		
3			x	x	x		x	x	x		x				x		
4			x	x		x	x	x	x		x			x			

5			x	x		x	x	x	x		x				x		
6			x	x		x	x	x	x		x						
7			x	x			x	x	x		x			x			
8			x	x	x	x		x	x			x			x	x	
9			x	x	x	x	x	x	x		x			x			
∑			**9**	**9**	**5**	**6**	**7**	**9**	**9**	**0**	**8**	**1**	**0**	**3**	**4**	**1**	

Table 9 - Interim Test UKC Coverage

6.6.4 An Analysis Against ATT&CK Tactics/Techniques

Within ATT&CK Enterprise there are 14 Tactics

These are

ID	Description
TA0001	Initial Access
TA0002	Execution
TA0003	Persistence
TA0004	Privilege Escalation
TA0005	Defense Evasion
TA0006	Credential Access
TA0007	Discovery
TA0008	Lateral Movement
TA0009	Collection
TA0010	Exfiltration
TA0011	Command & Control
TA0040	Impact
TA0042	Resource Development
TA0043	Reconnaissance

Table 10 - ATT&CK Tactics and Test Set

As above, we leave TA0042 and TA0043 out of scope.

	TA0001	TA0002	TA0003	TA0004	TA0005	TA0006	TA0007	TA0008	TA0009	TA0010	TA0011	TA0040	TA0042	TA0043
1	x	x					x				x			
2	x	x	x		x		x				x			
3	x	x			x		x				x			
4	x	x	x		x		x		x		x			
5	x	x	x		x		x				x			
6	x	x	x		x		x				x			
7	x	x			x		x		x		x			
8	x	x	x			x	x			x	x	x		
9	x	x	x		x		x		x		x			

Table 11 - Interim Test Tactic Coverage

From this we can see that some coverage of both UKC and ATT&CK Tactics are missing here.

In both UKC and ATT&CK we lack examples of Lateral Movement. This relates to Multi-Step type attacks. To a certain extent this reflects the content of many of the intelligence reports where the steps following initial access (IF and NP in UKC) phase are not well documented.

Similarly with Privilege Escalation, here these are referred to in the attack intelligence but have not been included in these sets. Some examples are available and can be added to address both these points.

The use of TA0010 is included but this is via a 'support' technique 'T1041 Exfiltration Over C2 Channel'. The actual detectable step technique is T1071 Application Layer Protocol representing a Command & Control Technique/Tactic. This suggests the relevant attack sequence needs to be adjusted. This is an example of where the attack sequencing from open reports still needs a measure of specialist expertise and interpretation.

Examples of note are found in APT28/APT29 attacks (shown in Appendix E) and these will now be added to the test set. These provide examples of Privilege Escalation steps and Lateral Movement within the organisation. Along with additional techniques.

The existing attack sequence Sandworm_001 will also be adjusted to replace T1071 with T1041 as shown above. T1071 is the detectable action and T1041 provides additional information about the purpose this action.

Lateral Movement is described by chaining sequential elements of attacks together. This detail is also shown on the attacks within Appendix E and will be explained in more detail later.

6.6.5 Adding Further Attacks to Complete Tactic and UKC Phase Coverage

After adding example APT28 and APT29 attacks we can now see that we have full coverage of across all ATT&CK Tactics and UKC phases.

The attacks now included are as follows (see Appendix E for details)

001 : APT28_001
002 : APT28_002
003 : APT28_003
004 : APT28_004
005 : APT29_001
006 : APT29_002
007 : APT29_003
008 : APT29_004
009 : APT32_001
010 : Lazarus_Group_001
011 : Lazarus_Group_002
012 : MuddyWater_001
013 : MuddyWater_002
014 : Mustang_Panda_001
015 : Sandworm_001
016 : Tropic_Trooper_001
017 : admin@338_001

With a coverage summary shown below

UKC Phases Data Coverage Summary

	IF-REC	IF-WEP	IF-DEL	IF-SEN	IF-EXP	IF-PER	IF-DEV	IF-C2C	NP-DIS	NP-PES	NP-EXE	NP-CAC	NP-LMV	AO-COL	AO-EXF	AO-TMA	AO-OBJ
Count	0	0	15	23	12	14	38	89	36	1	33	3	3	18	13	2	0

Figure 83 - Interim Test Set (UKC) Summary

ATT&CK Tactics Data Coverage Summary

	TA0043	TA0042	TA0001	TA0002	TA0003	TA0004	TA0005	TA0006	TA0007	TA0008	TA0009	TA0011	TA0010	TA0040
Count	0	0	20	42	12	1	38	5	35	3	19	91	9	2

Figure 84 - Interim Test Set (ATT&CK) Summary

6.6.6 Adding Further Attacks to Demonstrate Additional APT Groups

6.6.6.1 Justification for Full Test Set

Here we wish to demonstrate a reasonable coverage of additional groups as provided in previous works of a similar nature.

A good starting point is to be found in a developing suite of emulation instructions used by MITRE to evaluate various tools. This is the Centre for Threat Informed Defense Adversary Emulation Library. A background is to be found at (Miller et al., 2018) and a GitHub maintained and openly available set of descriptions can be found at (MITRE, n.d.-a). Presently eight APTs are studied, these are APT29, Carbanak, FIN6, FIN7, menuPass, Oilrig, Sandworm and Wizard Spider

The Unified Kill Chain described in (Pols, 2017) focusses on APT28 (already included).

The Mandiant Kill Chain is derived from an analysis of APT1 in (Mandiant, 2013).

In (Alshamrani et al., 2019) (a general survey of APTs) a number of APT behaviours are considered but in particular Deep Panda (also referenced in (Applebaum, 2016)) and Carabank campaigns are relevant here.

APT3, APT37 and APT41 are also found as exemplars in a number of papers.

The number of attacks used can be seen as similar to (Ehab Al-shaer, Ehab;Chu, 2017) and other papers.

Most research papers reviewed build models based on a similar number of key representative examples (and in fact often these APTs), in (Bahrami et al., 2019) a more comprehensive set of 40 APT group attacks (and campaigns) are analysed to form the justification of a categorisation model and we will aim for a similarly sized representative dataset. At this point the attacks can only be built manually and the time constraints required to construct these makes a manual review of all the references included in ATT&CK unrealistic.

In this work the test set was created manually. Some additional discussion on the automated generation of attack techniques sets is found in (Takahashi et al., 2020) (in particular section 3)

which lays out a strategy for future implementation. A general approach might take the ATT&CK group tactic/technique profile and build a set of 'randomly' generated attacks. Although this may work well to create a larger dataset that could be used to examine the effectiveness of approaches to matching 'observed' to 'recorded' technique sequences, several drawbacks seem evident.

- How can we constrain the synthesised attacks to retain the attack sequence structures related to a particular group (as mentioned previously the group profiles specifically do not include specific intelligence about sequence)
- Intelligence of techniques used by groups evident in reports is missing in the group profiles. There may be two likely reasons for this
    - The cost of the manual analysis has constrained the effort available to MITRE to be able to create the dataset
    - If a technique is achieved via the use of a tool and the technique is listed as a 'capability' of this tool then this may not be duplicated in the group profile.
- The reports included in the ATT&CK references have tended to provide relatively high-level intelligence. In (Takahashi et al., 2020) noted above it is interesting to note that the authors used a very small set of reports as these were the only ones available with the level of detail they required. For the ATT&CK references the Initial Access phase is normally described, but detail about subsequent actions and especially regarding lateral movement is limited. This need not be an issue in using this data to demonstrate the core principle proposed in this work.

6.6.6.2 Example APT Attacks Added

As a result of the above discussion a final set of 9 attacks was added:

These are:
001 : APT37_001
002 : APT3_001
003 : APT41_001
004 : APT41_002
005 : Carbanak_001
006 : FIN7_001
007 : OilRig_001
008 : WizardSpider_001
009 : menuPass_001

6.6.7 Test Set Summary

26 attacks have been codified (see also (Maidens, 2023)), shown below.

001 : APT28_001

002 : APT28_002

003 : APT28_003

004 : APT28_004

005 : APT29_001

006 : APT29_002

007 : APT29_003

008 : APT29_004

009 : APT32_001

010 : APT37_001

011 : APT3_001

012 : APT41_001

013 : APT41_002

014 : Carbanak_001

015 : FIN7_001

016 : Lazarus_Group_001

017 : Lazarus_Group_002

018 : MuddyWater_001

019 : MuddyWater_002

020 : Mustang_Panda_001

021 : OilRig_001

022 : Sandworm_001

023 : Tropic_Trooper_001

024 : WizardSpider_001

025 : admin@338_001

026 : menuPass_001

The overall different technique count is 97 of 576 Enterprise Techniques

Total observation (a step describing technique use in an attack) event count is 390

Here we have a high-level summary of the UKC and ATT&CK Tactic coverage. The counts show how many occurrences have been observed

UKC Phases Data Coverage Summary

	IF-REC	IF-WEP	IF-DEL	IF-SEN	IF-EXP	IF-PER	IF-DEV	IF-C2C	NP-DIS	NP-PES	NP-EXE	NP-CAC	NP-LMV	AO-COL	AO-EXF	AO-TMA	AO-OBJ
Count	0	0	28	34	14	22	88	133	51	2	97	6	5	28	21	3	0

Figure 85 - Final Test Set (UKC) Summary

ATT&CK Tactics Data Coverage Summary

	TA0043	TA0042	TA0001	TA0002	TA0003	TA0004	TA0005	TA0006	TA0007	TA0008	TA0009	TA0011	TA0010	TA0040
Count	0	0	31	74	21	5	89	6	52	5	29	131	20	4

Figure 86 - Final Test Set (ATT&CK) Summary

We can also show the distribution of technique coverage across the tactics

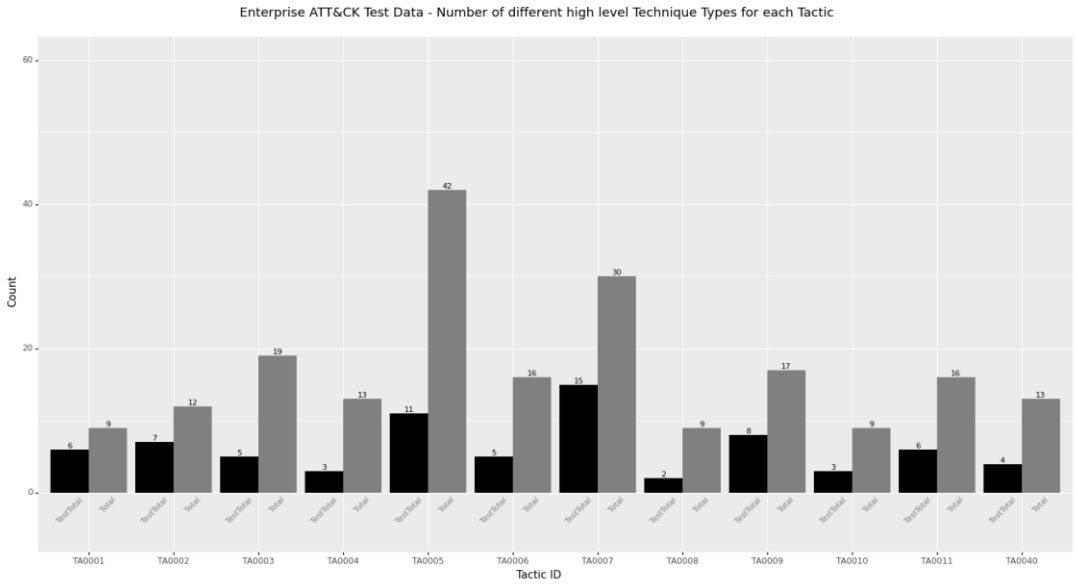


Figure 87 - Distribution of technique coverage in test data

6.6.8 Observations

While building this dataset several observations regarding the ATT&CK data set became apparent

- While working through the reports referenced by the ATT&CK group descriptions it was possible to identify techniques that were not included in the group descriptions
    - This is a constraint of the data; however, we should expect that any intelligence (no matter how comprehensive) can only include a subset of the real attacker's activities.

    - This may also be due to techniques being used by attackers through tools.
- Some of the groups only had limited source intelligence that could be used to understand attack structures.
    - This is to be expected and there may be a few reasons. Firstly, new groups or groups with limited activity will have limited intelligence available. Secondly, some groups will attract more interest and therefore more intelligence analysis than others.
- Most intelligence describes attackers outside of core 'Western' countries.
    - Given the source of this data this is to be expected. I have not fully investigated the availability of other open data sources (e.g., China, Russia)

6.7 Comparing the Initial and New Attack Model

This is the next step taken to define an attack sequence model as outlined in Introduction.

In this section the differences between the Initial and New Attack models are discussed. These models are as described previously in this chapter.

The key changes included in the New Attack model are as follows and are detailed further below

- The addition of a 'meta data model'.
- Addition of a 'Tinc' field. Intended to record timings of steps (from the start of the attack. This is currently unused but is for possible future use.
- Addition of a 'S/G' field to distinguish between functional steps taken by attackers and general qualities associated with techniques used.
- New assumptions on using the 'Pred' field in the new model.

6.7.1 The Meta Data Model

The motivation for the meta data model was to provide high level information across the population of multiple observed attacks that could also be used for additional trend analysis and categorisation across the data. This was not present in the Initial model so this objective could not have been supported.

All meta data recorded for the attacks used in this document are provided in the table below. This will be used to provide some brief illustrations of its use. The 'Sequence ID' column refers to the relevant attack sequence. A total of 31 attacks are documented here. 29 were used in the HMM analysis. The two remaining had limited intelligence documented (e.g TA551_001 had technical attack info but no additional context) so were suitable as meta data examples but the sequences were not used with the HMM model.

The model is described above (see The Meta Model) and is built on previous research proposals. A reminder of the elements (recorded as columns in the tables below) is given here:

Column / Dimension	***Description***
Attribution	Suggested APT that executed the attack if known
Init Access	Initial Access ATT&CK Technique/Vector
Origin	Country coordinating attack (if known)
Target	Country of target
Type	Target organisation type
Impact	The type of impact achieved (can be multiple outcomes
Vulns	Vulnerabilty/CVE references if applicable (CAPEC, CWE could be considered as well)
Related	A general field to allow association of attack sequences
Prec	Preceding Attack Sequence IDs to allowing linking of attacks
Ver	Version of schema used (to allow future changes of schema)
Date	Date of attack

Table 12 - Meta Data Dimension as Table Columns

All meta data recorded for the attacks used in this document are provided in the table below. This will be used to provide some brief illustrations of its use.

	Sequence ID	Attribution	Init Access	Origin	Target	Type	Impact	Vulns	Related	Prec	Ver	Date
	admin@338_001	admin@338	T1566.001 : Spearphishing Attachment	China	Hong Kong	Media	Exfiltration (confidentiality)	CVE-2012-0158		NA	1_0	2015
	Ajax_Security_Team_001	Ajax Security Team	T1566.001 : Spearphishing Attachment	Iran	Israel	Academic	Exfiltration (confidentiality)			NA	1_0	2015
	Andariel_001	Andariel	T1566.001 : Spearphishing Attachment	North Korea	South Korea	Security Researchers	Exfiltration (confidentiality) [UNCLEAR]			NA	1_0	2021
	APT1_001	APT1	T1566.001 : Spearphishing Attachment	China	United States	SCADA software engineering	Exfiltration (confidentiality)			NA	1_0	2012
	APT28_001	APT28	T1078 : Valid Accounts	Russia	United States	Government	Exfiltration (confidentiality)	CVE-2020-0688, CVE 2020-17144		NA	1_0	2021
	APT28_002	APT28	T1566.002 : Spearphishing Link	Russia	United States	Government	Exfiltration (confidentiality)			NA	1_0	2016
	APT28_003	APT28	T1566.002 : Spearphishing Link	Russia	United States	Government	Exfiltration (confidentiality)			NA	1_0	2016
	APT28_004	APT28	T1078 : Valid Accounts	Russia	United States	Government	Exfiltration (confidentiality)			APT28_003	1_0	2021
	APT29_001	APT29	T1566.001 : Spearphishing Attachment	Russia	United States	U.S.-based think tank	Exfiltration (confidentiality)	CVE-2021-36934		NA	1_0	2016
	APT29_002	APT29	T1566.001 : Spearphishing Attachment	Russia	United States	Ukraine based government department	Exfiltration (confidentiality)			NA	1_0	2018
	APT29_003	APT29	T1566.001 : Spearphishing Attachment	Russia	United States	U.S.-based think tank	Exfiltration (confidentiality)			NA	1_0	2016

	APT29_004	APT29	T1021.002 : <Remote Services>:SMB/Windows Admin Shares	Russia	United States	U.S.-based think tank	Exfiltration (confidentiality)			APT29_001	1_0	2016
	APT3_001	APT3	T1566.001 : Spearphishing Attachment	China	Hong Kong	Government Department	Exfiltration (confidentiality)	CVE-2015-3113		NA	1_0	2017
	APT32_001	APT32	T1566.001 : Spearphishing Attachment	Vietnam	Philippines	Government Department	Exfiltration (confidentiality)			NA	1_0	2017
	APT37_001	APT37	T1189 : Drive-by Compromise	North Korea	South Korea	Journalist	Exfiltration (confidentiality)			NA	1_0	2022
	APT38_001	APT38	T1566.002 : Spearphishing Link	North Korea	Brazil	Financial Institution	Damage (integrity)			NA	1_0	2018
	APT41_001	APT41	T1078 : Valid Accounts	China	United States	Government	Damage (integrity) Reputation			NA	1_0	2018
	APT41_002	APT41	T1195.002 : Compromise Software Supply Chain	China	United States	Healthcare	Exfiltration (confidentiality)			APT41_001	1_0	2018
	Carbanak_001	Carbanak	T1566.001 : Spearphishing Attachment	Ukraine	Russia	Banking	Damage (integrity and availability), Reputation (external to the target)	CVE-2012-0158, CVE-2013-3906, CVE- 2014-1761, CVE-2013-3660		NA	1_0	2014
	FIN7_001	FIN7	T1566.001 : Spearphishing Attachment	Russia	United States	Retail chain	Damage (integrity and availability), Reputation (external to the target)			NA	1_0	2017
	Lazarus_Group_001	Lazarus Group	T1566.001 : Spearphishing Attachment	North Korea	Turkey	Banking	Exfiltration (confidentiality)	CVE-2018-4878		NA	1_0	2018
	Lazarus_Group_002	Lazarus Group	T1566.001 : Spearphishing Attachment	North Korea	Turkey	Banking	Exfiltration (confidentiality)	CVE-2018-4878		NA	1_0	2018

	menuPass_001	menuPass	T1190 : Exploit Public Facing Application	China	Japan	Manufacturing	Exfiltration (confidentiality)	"CVE-2019-11510", "CVE-2019-11539"		NA	1_0	2019
	MuddyWater_001	MuddyWater	T1566.001 : Spearphishing Attachment	Iran	Pakistan	Defense	Exfiltration (confidentiality)			NA	1_0	2018
	MuddyWater_002	MuddyWater	T1566.001 : Spearphishing Attachment	Iran	Turkey	Government	Exfiltration (confidentiality)			NA	1_0	2018
	Mustang_Panda_001	Mustang Panda	T1566.001 : Spearphishing Attachment	China	Hong Kong	Catholic Church	Exfiltration (confidentiality)			NA	1_0	2020
	OilRig_001	OilRig	T1566.003 : Phishing: Spearphishing via Service	Iran	United States	Oil and Gas	Exfiltration (confidentiality)			NA	1_0	2019
	Sandworm_001	Sandworm Team	T1566.001 : Spearphishing Attachment	Russia	Ukraine	Government	Exfiltration (confidentiality), Damage (integrity and availability)	CVE-2014-4114		NA	1_0	2014
	TA551_001	TA551	T1566.001 : Spearphishing Attachment	Unknown	Unknown	Unknown	Unknown	CVE-2012-0158		NA	1_0	2020
	Tropic_Trooper_001	Tropic Trooper	T1566.001 : Spearphishing Attachment	China	Taiwan	Government	Exfiltration (confidentiality)	CVE-2012-0158		NA	1_0	2020
	WizardSpider_001	WizardSpider	T1566.001 : Spearphishing Attachment	Russia	United States	Hospital	Damage (integrity and availability)			NA	1_0	2022

Some brief examples of how this data can be used is provided here (and were not possible using the Initial model)

Rows can be selected by any combination of columns. Such as:

- By using Origin, Target and Type it is possible to look at trends of behaviours between nation states. This can be also refined to focus on attacks by specific APTs.
    - E.g. Looking at (assumed) Origin = Iran reveals {United States, Pakistan, Turkey, Israel} as Targets observed
- Using the Date column combined with additional columns, yearly trends can be explored.
    - E.g. Year = 2020 provides rows {Mustang_Panda_001, TA551_001, Tropic_Trooper_001}
- Using the CVE APT use of vulnerabilities can be observed
    - E.g. Vulns = CVE-2012-0158 provides { admin@338 (China), Carbanak (Ukraine), Tropic Trooper (China), TA551 (Unknown) }

6.7.1.1 The Meta Data 'Preceded By' Field

As mentioned previously (see The Meta Model) an attack sequence begins with either initial access or following pivot/lateral movement. It ends at access to a targeted asset or pivot/lateral movement.

The 'Precede By' field allows linking of attack sequences to provide descriptions of longer attacks.

Examples above include:

- APT28_004 following APT28_003. In this case a number of years have passed while the persistent threat was not discovered and moved through the target's systems.
- APT41_002 following APT41_001. In this case representing the steps in a supply chain attack.
- APT29_004 following APT29_001. In this case an attack pivoting through deployment of malware at initial access. APT29_001 will receive the data collected (represented as a sequence of techniques within this sequence) as a result of lateral movement and tool transfer in APT29_004 and then exfiltrate.

Some examples of how sequences may be logically linked is shown below:

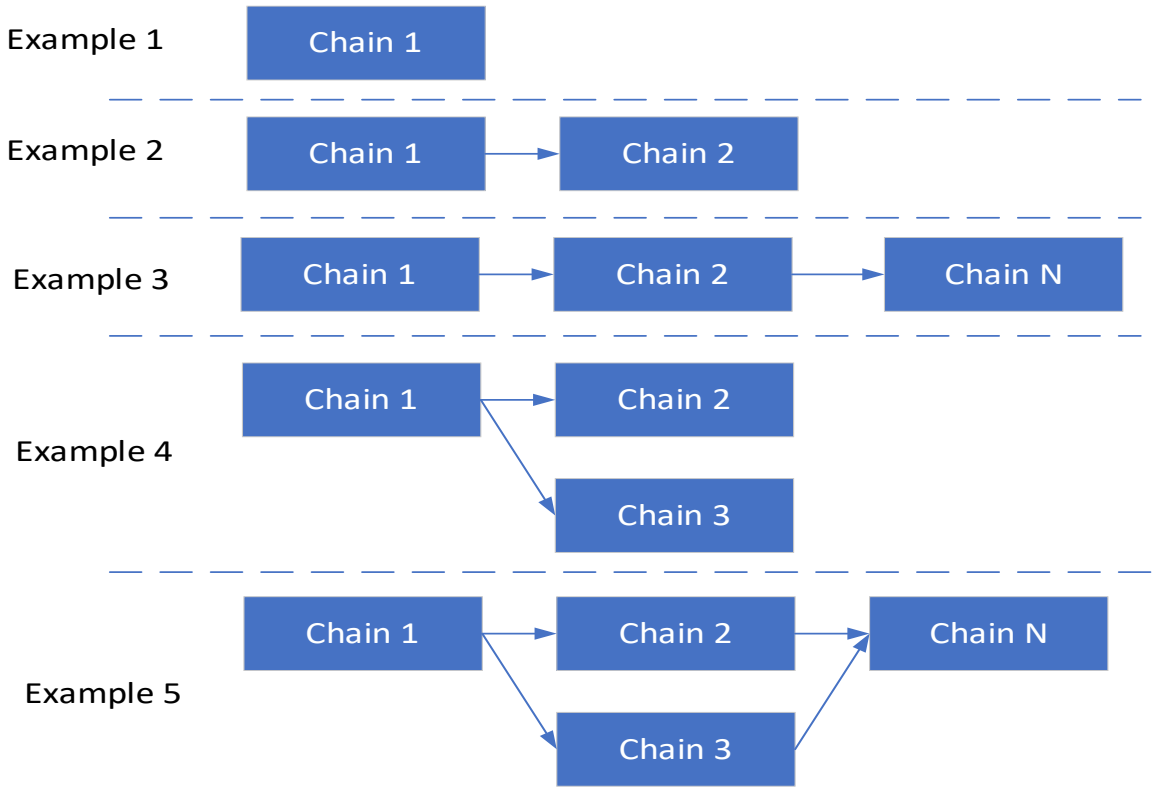


Figure 88 - Examples of Linked Sequences

6.7.2 Addition of a 'Tinc' field

The 'Tinc' field is intended to provide timings for the start of each step (offset from the start of the attack sequence).

The 'Tinc' field is not populated in the model described here but has been included for future use. This field is not populated because of the nature of the threat intelligence reporting generally available. This will be described in a bit more detail below.

As noted previously (in Sequence Comparison), (Studer & Ritschard, 2016) provides a good summary to general approaches in identifying distances between sequences.

The following aspects are identified as important.

- "experienced states—the distinct elements of the alphabet present in the sequence;"
- "distribution—the within-sequence state distribution (total time);"
- "timing—the age or date at which each state appears;"
- "duration—the spell lengths in the distinct successive states;"
- "sequencing—the order of the distinct successive states."

Without a timing element only the first and last aspects are addressed.

- Experienced states – Addressed by the Tactic and Technique fields.
- Sequencing – The steps in the attack sequence.

The addition of the 'Tinc' field allows the introduction of future content that can be used to address the middle three aspects.

- Distribution – Total time spent by the attacker in each Tactic or Technique.
- Timing – Offset time of each step.
- Duration – The time spent at each step.

As mentioned above the content developed and used in the demonstration of the model in this thesis could not contain timings as these are not present in the available intelligence reports. However, the potential for this intelligence to be added has been put in the model (see also Future Work).

6.7.3 Addition of a 'S/G' field

While working through the attack reports and representing the attacks as streams of tactics and techniques it became clear that some techniques (and information in the reports) did not actually describe functional steps taken by the attackers but rather qualities of the functional steps taken. The initial model did not distinguish between the functional and additional information technique use.

An example is provided in the techniques associated with sending and receiving command and control messages. For instance, from admin@338_001 (see also The Refined Attack Sequence Model) we have the extract:

	Sequence_ID	***admin@338_001***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
4	TA0011 : Command & Control	T1071.001 Application Layer Protocol : Web Protocols	3		G	IF-C2C	This technique to provides more detail on step below
5	TA0011 : Command & Control	T1102.002 Web Service: Bidirectional Communication	3		S	IF-C2C	[1] Initial installation connects to C&C

Table 13 - Attack sequence extract showing use of S/G field

The report provides information about the point in the attack sequence when the messages were sent, but also information about the methods used to send and receive messages. Using the S/G flag the additional information can be recorded while retaining the logical sequence of the actions taken by the attacker. The 'S' value indicates the actual functional act of sending a message. The 'G' value indicates additional supporting information about how that message was sent.

This supporting information can potentially be used to support finer grained analysis of observed sequences. For instance, if a C&C sequence is observed matching a number of known attack sequences, the exact supporting techniques (e.g., data encoding approaches) used to implement the messages could help distinguish between them.

This does not appear to be a distinction held in the ATT&CK knowledge base. Within the Command & Control Tactic it is often possible to identify multiple techniques that are required to fully describe (based on information in the reports) a single functional step taken by an attacker. In the Future Work section, a suggestion is made to rationalise this (see Future Work 'Use Data to Motivate a More Formal Analysis of the ATT&CK Knowledge Base Structure').

A couple of small attack sequence fragments are shown below to illustrate the logical use of this field. These are from attacks by APT1 and Tropic Trooper (see also Appendix C).

- First functional step - Both use a spearphishing attachment as the initial access technique (T1566.001).
- Second functional step - The next action taken is through the execution of the malicious attachment (T1204.002).
- In both attacks the malicious attachment is disguised look like a normal PDF file (T1036). This is through preparation by the attacker and is not an action within the attack. This is therefore marked with a G.
- In the second attack the attacker has also prepared the code using obfuscation techniques (T1207). This is also marked with a G.

So, in this case if an observer detected the techniques T1566.001 followed by T1204.002 then either attack APT1_001 or Tropic_Trooper_001 could be deemed to be possible matches. If subsequent analysis noted that the malicious attachment used the appropriate obfuscation techniques then this can narrow the possible matches to just Tropic_Trooper_001.

	Sequence_ID	*APT1_001*		*Ver*	*0.1*		
ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S		

2	**TA0005 : Defense Evasion**	**T1036 : Masquerading**	1		G		Some APT1 actors have gone to the trouble of making the malicious software inside their ZIP files look like benign Adobe PDF files.
3	**TA0002 : Execution**	**T1204.002 : User Execution: Malicious File**	1		S		APT1 establishes a foothold once email recipients open a malicious file and a backdoor is subsequently installed. This file is actually a dropper for a custom APT1 backdoor that we call WEBC2-QBP The steps describing exactly how the malware is installed is not available. We will assume that a2 WebC2 backdoor is installed and this then downloads a standard backdoor as this is broadly referred to broadly in other reports around this time

Table 14 - APT1_001 example attack sequence extract

	Sequence_ID	***Tropic_Trooper_001***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158
2	**TA0005 : Defense Evasion**	**T1036 : Masquerading**	1		G	IF-DEV	The delivery document uses the XLSX extension typically used by OpenXML documents, but the file itself is actually an OLE (XLS) document
3	**TA0005 : Defense Evasion**	**T1027 : Obfuscated Files or Information**	1		G	IF-DEV	… which stores XLSX ciphertext and the information needed for decryption in an OLE document
4	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158

Table 15 - Tropic_Trooper_001 example attack sequence extract

6.7.4 New assumptions on using the 'Pred' field in the existing model

In both the Initial and New Attack Model, the Pred field is used to order the steps within the attack sequence. The Pred field references the id of the previous step of the attack. Although logically the same function is performed in the New Model, the approach used in recording attacks has been adjusted:

Firstly, in the original example attack built (Example for admin@338) the multiple Discovery type techniques observed (Steps 8 to 13) were shown as following a single command script execution step (Step 7). Step 14 then following on from this group. This was intended to document a script being run with multiple commands in it. An extract is shown below:

ID	***Tactic***	***Technique***	***Pred***
7	TA0002 : Execution	T1059.003 Command and Scripting Interpreter: Windows Command Shell	6
8	TA007 Discovery	T1083 File and Directory Discovery	7

9	TA007 Discovery	T1082 System Information Discovery	7
10	TA007 Discovery	T1016 System Network Configuration Discovery	7
11	TA007 Discovery	T1007 System Service Discovery	7
12	TA007 Discovery	T1069.001 Permission Groups Discovery: Local Groups	7
13	TA007 Discovery	T1049 System Network Connections Discovery	7
14	TA0011 : Command & Control	T1105 Ingress Tool Transfer	8-13

In the new model the usage was adjusted (but the model was not logically changed) as shown in the extract below (see also The Refined Attack Sequence Model). Instead of showing the Discovery techniques as 'a group' they are now shown as individual steps. This is to reflect the exact order of the steps taken (via scripts or otherwise). The model still supports the previous approach but was not used for the demonstrations here. This simplification was to allow direct matching of sequences. In Future Work (see Future Work – 'Further Formalise the Attack Sequence Model') a suggestion to consider matching approaches for sequences containing 'groups' of techniques (where multiple orderings of group elements may be considered) is suggested.

ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***
7	TA0002 : Execution	T1059.003 Command and Scripting Interpreter: Windows Command Shell	6		S
8	TA007 Discovery	T1083 File and Directory Discovery	7		S
9	TA007 Discovery	T1082 System Information Discovery	8		S
10	TA007 Discovery	T1016 System Network Configuration Discovery	9		S
11	TA007 Discovery	T1007 System Service Discovery	10		S
12	TA007 Discovery	T1069.001 Permission Groups Discovery: Local Groups	11		S
13	TA007 Discovery	T1049 System Network Connections Discovery	12		S
14	TA0011 : Command & Control	T1071.001 Application Layer Protocol : Web Protocols	13		S

Secondly, the Pred field was used to support the 'G' elements in the attack. Shown the addition of the S/G field (see previous section). Using the same example as above (see Addition of a 'S/G' field). Both ID 14 and 15 note ID 13 as the predecessor.

In the demonstrations below (see Results) the functional attack sequences used by the attacker are built from the S steps. The G elements then provide additional context about the qualitative nature of the S steps (when the intelligence is available).

ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***
13	TA007 Discovery	T1049 System Network Connections Discovery	12		S

14	TA0011 : Command & Control	T1071.001 Application Layer Protocol : Web Protocols	13		G
15	TA0011 : Command & Control	T1102.002 Web Service: Bidirectional Communication	13		S

Here the attacker is sending data, found through Discovery actions, back to a C&C server. The act of sending it back is shown is Step 15. Step14 (a G step) provides additional information about how the step was implemented (T1102.002 provides an incomplete description). In the previous model if Step 14 was included it would be unclear whether there were actually two command and control messages being sent. If Step 14 was omitted then clearly there would be a reduction in intelligence available for subsequent analysis.

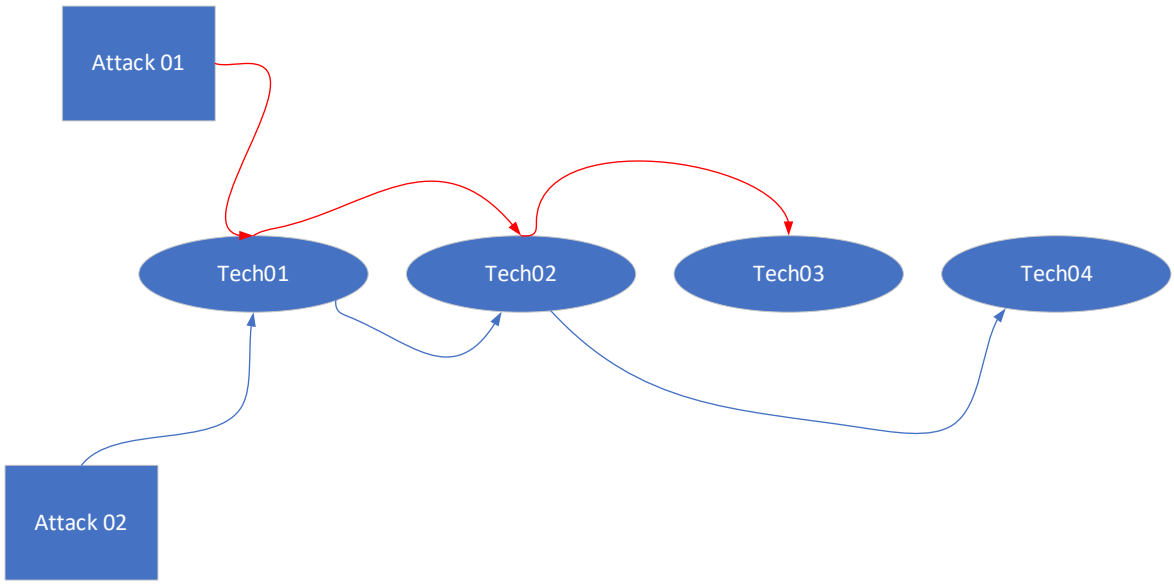
6.7.5 Small Note on Implementation of Graph Model

One optional approach to implementing a database of the sequences is using a graph model.

This is briefly described in Loading the New Attack Model into an Example Database.

There is no special reason for using a graph-like implementation however the node and edge model can offer some implementation advantages depending on the purpose of the application.

A toy example of technique sequences for two dummy attacks is given below



By implementing the knowledge in this manner, we can then understand metrics on transitions between techniques simply by reading off the count of edges between the two nodes.

For instance, here we can see two edges representing instances of transitions between Tech01 and Tech02.

Chapter 7 Results

7.1 Introduction

In this section examples are given to show how a knowledge base of attacks described as sequences of ATT&CK Tactics and Techniques, can provide an additional source of cyber threat intelligence to complement the existing MITRE ATT&CK Knowledge Base.

To demonstrate this, cyber-attacks described in open-source intelligence are described as sequences of ATT&CK tactics and techniques using the model developed in Building a Model. These attacks are then used to illustrate the examples shown below.

Four examples are illustrated

- In Using the Attack Model – LCSS Fragment Matching an example is given of how matching an observed sequence of attack techniques to a set of known APT attack sequences can provide an increase in the intelligence obtained over that gained from simple lists of techniques used by APTs. This example uses Longest Common Sub String (LCSS) to match sequences.
- In Using the Attack Model – Hidden Markov Model an example is given of how an observed sequence of techniques may be used to predict what ATT&CK tactics are being used by the attacker. This example uses a Hidden Markov Model to predict the tactics.
- In Using the Attack Model – Markov Model an example is given of how an observed sequence of techniques may be used to predict the next technique likely to be used. This example uses a Markov Model to predict the tactics.
- In Using the Attack Model – Unified Kill Chain the attack sequences created are examined to try and provide insight into how the ATT&CK Tactics used map onto a kill chain model. This example uses the Unified Kill Chain model.

7.2 Using the Attack Model – LCSS Fragment Matching

7.2.1 Introduction

This is the next example as outlined in Introduction.

The objective in this section is to provide an illustration of how a sequence of detected Techniques/Tactics (Attack Fragment) representing a potential attack, can be compared with a set of known Attack Sequences recorded using the model developed above (A New Attack Model).

The original results found in Motivating Example are compared with a ranked order of how well the Attack Fragment matches each of the Attack Sequences within the Knowledge Base. This is to show that additional intelligence is obtained. The matching approach used is Longest Common Subsequence (LCSS).

This intelligence could then be used to provide guidance to analysts to help decide on potential courses of action and expected next attack steps.

In the section below:

- The approach is described in Approach
- The results of three tests are described in Results
- A summary of conclusions is discussed in Conclusions

7.2.2 Approach

For this example, the attack sequence examples developed during the model analysis (see Test Set Summary) were loaded into a basic graph database. There are many ways that the data could have been persisted for processing but the graph database approach was used here to also illustrate the model shown in Loading the New Attack Model into an Example Database. All code and the graph database used may be found at (Maidens, 2023).

A summary of the approach is as follows:

- Place all known attack sequences (modelled as in A New Attack Model) in a database (a graph database in this case).
- Record an observed attack sequence (in this case the attack sequence used was as shown in A Motivating Simple Initial Example).

- Compare the observed attack sequence with **each** of the known attack sequences and calculate metrics show how well they match.
    - Calculate LCSS between Observed Attack sequence and **this** Known Attack sequence
    - Replace each sub-technique in Observed and **this** Known Attack sequence with their parent techniques and calculate LCSS between results
    - Calculate LCSS between Observed Attack Tactic sequence and **this** Known Attack Tactic sequence
- Rank how well the observed sequence matches each of the known attack sequence using the metrics calculated.

7.2.2.1 Ranking Approach

The ranking will be achieved using a hierarchical set of values obtained through the use of LCSS.

As described in the approach above the observed attack sequence with **each** of the known attack sequences and three metrics calculated to understand how well they match. The results are then ordered by

LCCS between Observed Attack sequence and Known Attack sequence

Then - LCSS between Observed and Known Attack sequence with each sub-technique replaced with their parent techniques

Then - LCCS between Observed Attack Tactic sequence and Known Attack Tactic sequence

A record is also kept of the SimHash distance between the Observed Attack sequence and the Known Attack sequence for comparison (this can be seen in the table below).

Simhash was developed by Google and is used for identifying similar content. “Simhash works by dividing the input into smaller chunks, called “features,” and then generating a hash of each feature. These hashes are then combined to produce the final hash for the input.” (Otten, 2023)

There are other approaches that compare favourably with LCSS. The use here is included primarily as demonstration of the value of the data. However, LCSS is valid approach that is frequently used. From (Vlachos et al., 2002) we have relevant advantages for its use here:

- LCSS can deal with sequences with different lengths – This is important in this application as we cannot guarantee the completeness of the detected sequences
- LCSS can deal with outliers – Again this important when any detection system is likely to produce anomalous positives.

Further development of the approach here is also suggested in Future Work.

Other sequence matching approaches were considered. (Cheng et al., 2011) uses correlated security alerts to build a likely attack sequence through comparison with actual attack sequences. Another is given in (Vlachos et al., 2002) where the authors define 'non-metric similarity functions' based on the Longest Common Subsequence (LCSS) (for trajectories in two and three dimensional space). The (Vlachos et al., 2002) approach is resistant to noise (which is something we must assume in any theoretical detection system here) and provides an 'intuitive notion of similarity between trajectories by giving more weight to the similar portions of the sequences'.

7.2.2.2 LCSS Alphabet definition in this application

It is also worth clarifying the alphabet to be used here as well. Some efficiency improvements to LCSS implementations are sensitive to both string and alphabet sizes.

As clarified below the input 'strings' will be ordered sequences of either Tactics or Techniques

In this implementation each of the Techniques/Tactics (there are currently 668 possible techniques) is allocated a numeric value stored as a Unicode value (the natural char type in Python).

The technique/tactic sequences are then presented as a stream of Unicode characters and compared in this way.

7.2.3 Results

An initial review of the new attack model was conducted to show that the intelligence provided by the attack sequences can improve on the results shown in the Initial example attempt (Motivating Example) by providing a ranked order of matching.

Within the knowledge base we have the example attacks from admin@338, APT32, Lazarus Group, MuddyWater, Mustang Panda, Sandworm Team and Tropic Trooper (plus additional attacks as described at Test Set Summary).

The original test attack sequence ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049'] was then compared with the given example attacks using the LCSS approach described above.

Output from the tests are shown below and interpretation is given in the Conclusions below.

First Test Run

The Test Attack Techniques are ['T1566.001', 'T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1007', 'T1069.001', 'T1049']

Attack	Technique LCS	Main Tech LCS	Tactic LCS	Distance [0,1] 1 is closest
admin@338_001	9	9	9	0.7276
MuddyWater_001	6	7	7	0.5547
MuddyWater_002	6	7	7	0.5164
Lazarus_Group_002	5	5	6	0.4303
APT32_001	4	5	5	0.2910
Mustang_Panda_001	4	4	6	0.3059
FIN7_001	4	4	5	0.2615
APT29_001	3	4	5	0.2673
Lazarus_Group_001	3	3	5	0.2582
Sandworm_001	3	3	5	0.2673
APT29_003	3	3	4	0.2673
APT29_002	3	3	3	0.2236
Tropic_Trooper_001	2	3	6	0.1455
WizardSpider_001	2	3	5	0.1491
APT37_001	2	3	3	0.1571
APT41_002	2	2	4	0.2010
Carbanak_001	2	2	4	0.1617
APT3_001	1	4	6	0.0808
OilRig_001	1	3	5	0.0711
menuPass_001	1	1	4	0.0765
APT28_001	1	1	2	0.1260
APT28_004	0	1	3	0.0000
APT28_002	0	1	1	0.0000

APT28_003	0	1	1	0.0000
APT41_001	0	0	1	0.0000
APT29_004	0	0	0	0.0000

Second Test Run

The Test Attack Techniques are ['T1204.002', 'T1059.003', 'T1083', 'T1082', 'T1016', 'T1069.001', 'T1049']

Attack	**Technique LCS**	**Main Tech LCS**	**Tactic LCS**	**Distance [0,1] 1 is closest**
admin@338_001	7	7	7	0.6417
MuddyWater_001	5	6	6	0.5241
MuddyWater_002	5	6	6	0.4880
Lazarus_Group_002	4	4	5	0.3904
APT32_001	3	4	4	0.2474
Mustang_Panda_001	3	3	5	0.2601
FIN7_001	3	3	4	0.2224
APT29_001	2	3	4	0.2020
Lazarus_Group_001	2	2	4	0.1952
Sandworm_001	2	2	4	0.1952
APT29_003	2	2	3	0.2020
APT41_002	2	2	3	0.2279
APT29_002	2	2	2	0.1690
APT37_001	2	2	2	0.1782
APT3_001	1	3	5	0.0917
Tropic_Trooper_001	1	2	5	0.0825
WizardSpider_001	1	2	4	0.0825

OilRig_001	1	2	4	0.0788
menuPass_001	1	1	3	0.0867
Carbanak_001	1	1	3	0.0917
APT28_001	1	1	1	0.1336
APT28_004	0	1	2	0.0000
APT28_002	0	0	0	0.0000
APT28_003	0	0	0	0.0000
APT29_004	0	0	0	0.0000
APT41_001	0	0	0	0.0000

Third Test Run

The Test Attack Techniques are ['T1204.002']

Attack	**Technique LCS**	**Main Tech LCS**	**Tactic LCS**	**Distance [0,1] 1 is closest**
Lazarus_Group_001	1	1	1	0.2582
admin@338_001	1	1	1	0.2425
Lazarus_Group_002	1	1	1	0.2582
APT32_001	1	1	1	0.2182
MuddyWater_001	1	1	1	0.2774
MuddyWater_002	1	1	1	0.2582
Mustang_Panda_001	1	1	1	0.2294
Sandworm_001	1	1	1	0.2582
Tropic_Trooper_001	1	1	1	0.2182
APT29_001	1	1	1	0.2673
APT29_002	1	1	1	0.2236

APT29_003	1	1	1	0.2673
Carbanak_001	1	1	1	0.2425
APT37_001	1	1	1	0.2357
WizardSpider_001	1	1	1	0.2182
FIN7_001	1	1	1	0.1961
APT3_001	0	1	1	0.0000
APT28_001	0	0	1	0.0000
APT28_004	0	0	1	0.0000
APT41_002	0	0	1	0.0000
menuPass_001	0	0	1	0.0000
OilRig_001	0	0	1	0.0000
APT28_002	0	0	0	0.0000
APT28_003	0	0	0	0.0000
APT29_004	0	0	0	0.0000
APT41_001	0	0	0	0.0000

7.2.4 Conclusions

Originally, (in Motivating Example) we found that the Test Attack (shown above) most likely matched the technique set recorded for admin@338 by simply counting the matching techniques. We also found that the other groups (APT32, Lazarus Group, MuddyWater, Mustang Panda, Sandworm Team and Tropic Trooper) all matched on the same set of seven techniques. So, we were not able to distinguish between these groups.

For the first test result, we are now matching the attack set as a sequence to several attack sequences derived from known attack intelligence reports. The intention is not to assume attribution to a particular APT but simply find closest attack sequences.

The table of results is ranked in order of the columns. The attacks discussed in Motivating Example are highlighted in grey.

As can be seen admin@338_001 is still detected as the most likely match. But now we can also provide a ranking of matching against the other attacks to provide a prioritised list of investigation (previously we could not distinguish between results).

Oilrig was previously identified as a good match but in this case the Oilrig example attack included in the knowledge base does not match the detected sequence well at all. This is an example of distinguishing between matching known attacks and APT attribution. An APT may have a set of techniques that it has been observed using (and therefore recorded in MITRE ATT&CK) but this is no indication that these were all used together in attacks. These techniques may have been observed over a long period of time in completely separate attacks with the APT evolving their approach over that period.

In the second test a smaller set of observed techniques is presented (in the same order). This has still resulted in a similar ranking at the top of the table. Tropic_Trooper_001 is now a less good match as matches that were previously present are now missing.

Although not used to order the table the Simhash column (Distance[0,1]) provides additional input that could be used to distinguish between rows with matching LCSS values (e.g. MuddyWater_001 and MuddyWater_002).

In general, the Simhash ranking matches the LCSS approach quite well. However, for APT32_001 and Mustang_Panda_001 Simhash provides a slightly different ordering. The intention here was to use the LCSS approach as this is tailored to the specific known qualities of the meaning of the data in this context (i.e. Simhash is tailored to general 'documents' and not cyber-attack sequences).

In the third test consideration is given about the general meaning of the matching. Detected sequences may contain varying levels of information. Large amounts of 'noise' may result in a match to an irrelevant sequence. Low amounts of information may similarly skew matching results especially where the matching index takes the lengths of the observed and known example strings into account (e.g., if we have four observed techniques and this observed sequence matches two examples of attack sequence for all four observed techniques. Do we consider the length difference between the observed and example sequences in the measure of closeness?).

This test provides an extreme example of matching on just one observation in the sequence. Here the result is essentially just a list of the example attacks with one match (in the order that they are read). The Simhash index also includes consideration of the difference in string lengths, which would be of doubtful value in this case.

This suggests a need to understand the (average) behaviour of a detection system when considering a matching approach.

7.3 Using the Attack Model – Hidden Markov Model

7.3.1 Introduction

This is the next example as outlined in Introduction.

The attack sequences stored in the model described above (A New Attack Model) can be used to understand the transitions between various Tactic and Techniques described in the attacks.

A common assumption in research is that cyber-attacks may be considered as Markov processes (see also Markov Models) (e.g. (Dass et al., 2021), (Ye et al., 2004),(Chadza et al., 2020)). This section treats the techniques as observations and the tactics as the hidden states and builds a Hidden Markov model (HMM) to derive attack tactic sequences from observed technique streams.

This useful because not all techniques are directly associated with a single tactic and knowledge of most likely tactic provides knowledge of potential attacker's intention. A sample output from the ATT&CK database used in this work is shown below:

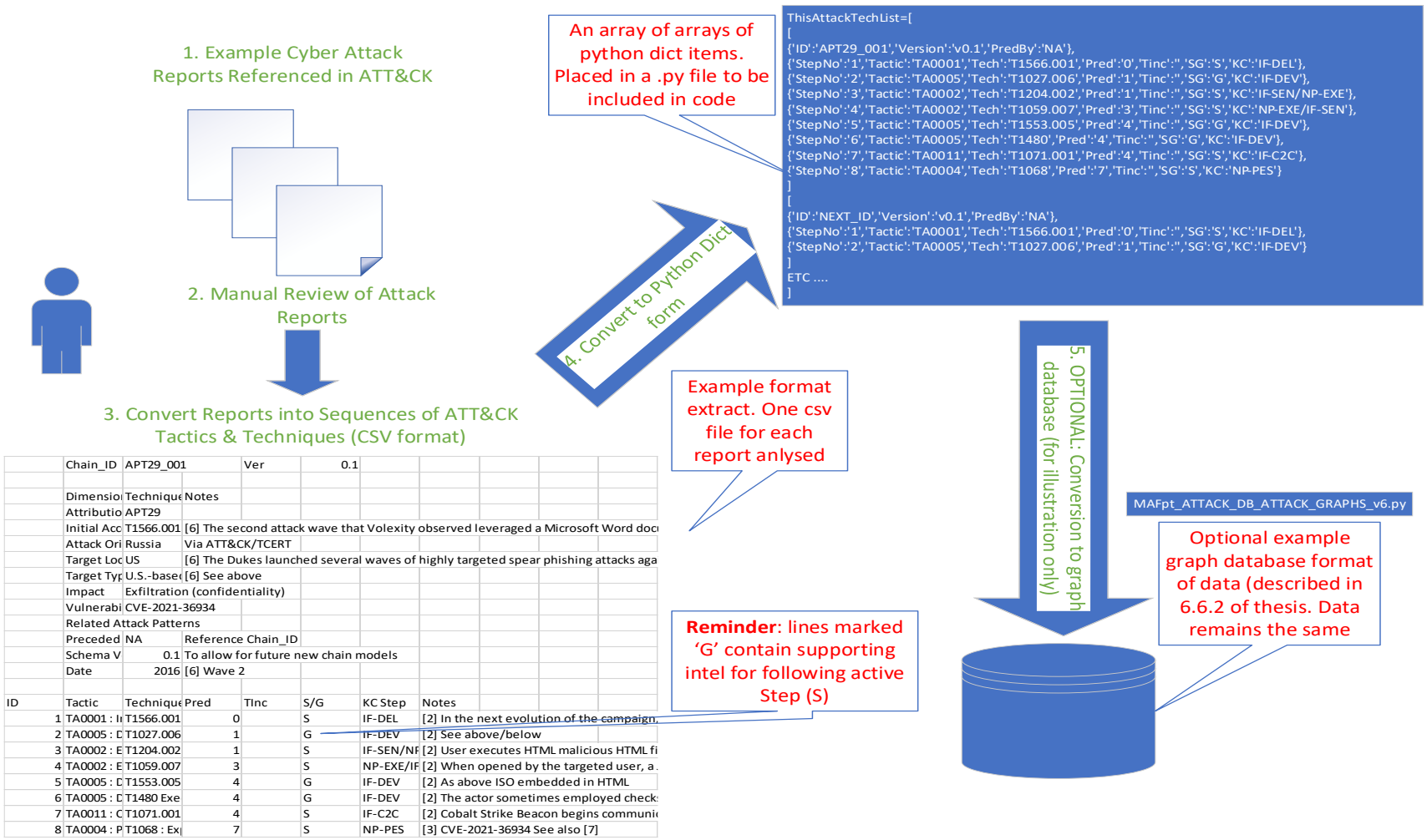
> There are 306 techniques that use 1 tactics
> There are 51 techniques that use 2 tactics
> There are 11 techniques that use 3 tactics
> There are 4 techniques that use 4 tactics

In the section below:

- The approach is described in Approach
- The results of three tests are described in Results
- A summary of conclusions is discussed in Conclusions

7.3.2 Approach

7.3.2.1 Data Preparation



	Chain_ID	APT29_001		Ver	0.1					
	Dimensio	Technique	Notes							
	Attributio	APT29								
	Initial Acc	T1566.001	[6] The second attack wave that Volexity observed leveraged a Microsoft Word doc							
	Attack Ori	Russia	Via ATT&CK/TCERT							
	Target Loc	US	[6] The Dukes launched several waves of highly targeted spear phishing attacks aga							
	Target Typ	U.S.-base	[6] See above							
	Impact	Exfiltration (confidentiality)								
	Vulnerabi	CVE-2021-36934								
	Related Attack Patterns									
	Preceded	NA	Reference Chain_ID							
	Schema V	0.1	To allow for future new chain models							
	Date	2016	[6] Wave 2							
ID	Tactic	Technique	Pred	Tinc	S/G	KC Step	Notes			
1	TA0001 : I	T1566.001	0		S	IF-DEL	[2] In the next evolution of the campaign,			
2	TA0005 : D	T1027.006	1		G	IF-DEV	[2] See above/below			
3	TA0002 : E	T1204.002	1		S	IF-SEN/NP	[2] User executes HTML malicious HTML fi			
4	TA0002 : E	T1059.007	3		S	NP-EXE/IF	[2] When opened by the targeted user, a			
5	TA0005 : D	T1553.005	4		G	IF-DEV	[2] As above ISO embedded in HTML			
6	TA0005 : D	T1480 Exe	4		G	IF-DEV	[2] The actor sometimes employed check			
7	TA0011 : C	T1071.001	4		S	IF-C2C	[2] Cobalt Strike Beacon begins communi			
8	TA0004 : P	T1068 : Ex	7		S	NP-PES	[3] CVE-2021-36934 See also [7]			

Figure 89 - Data Preparation

A summary of the steps illustrated above

1. The APTs and associated reports selected from within ATT&CK are described in Loading a Representative Data Set
2. The reports were manually analysed to extract example attacks. Reports often contain details of a range of attacks and in disordered sections so manual intervention required to extract sequences within the reports.
3. The attack sequence observed was then placed in a CSV file (one for each attack analysed). The general format is illustrated in the above illustration but is detailed in A New Attack Model.
4. In the first instance the CSVs are converted to an array of Python dict arrays (in a .py source file). In summary there is an array where each item represents an attack sequence. Each attack sequence (itself an array of dicts) contains a list of steps taken by the attacker. Each step is represented in a Python dict (eg {'**StepNo**':'1','**Tactic**':'TA0001','**Tech**':'T1566.001','**Pred**':'0','**Tinc**':'','**SG**':'S','**KC**':'IF-DEL'}).
    - It should be noted that the first dict in each attack sequence represents this attack's meta data (e.g. {'**ID**':'APT29_001','**Version**':'v0.1','**PredBy**':'NA'}).
    - For these examples only a subset of the possible metadata was loaded to provide a general example of approach.
    - The meta and the attack sequence dict fields are as detailed in the A New Attack Model.
5. The python dict data structures contain all the sequence data included in the original csv files (as in Step 3 above). This can be used to populate additional databases depending on processing requirements. A graph database example was built. The structure of this database is described in Loading the New Attack Model into an Example Database. The code to load and process this instantiation of the data is found at (Maidens, 2023).

7.3.2.2 Approach Summary

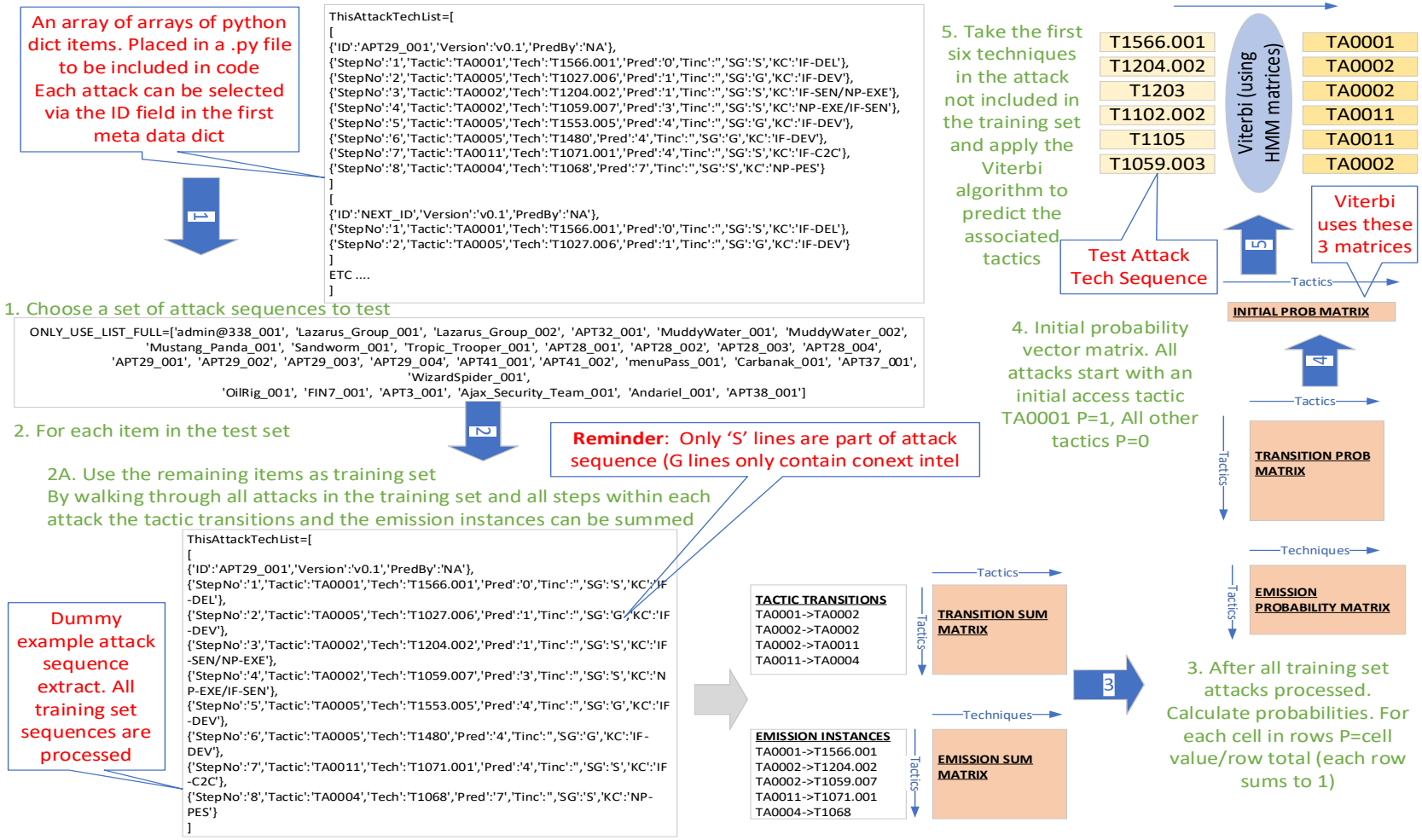


Figure 90 - Creating HMM matrices

A summary of the approach is provided here (illustrated above) with supporting detail below:

- Code and data used for this test may be found at (Maidens, 2023).
    - The data preparation is described above (Data Preparation see also Step 1 above).
- The set of attack sequence examples is taken as input data.
    - Details of the attack sequences used within the tests are given in the results section below
- For **each individual attack** in the example set (this is called a Leave One Out approach) – see also Step 2 above.
    - The individual attack was taken out of the whole set.
    - The rest of the example set was used as the training set
    - The training set was used to create HMM matrices (as described below, see also Step 2, 3 and 4 illustrated above))
    - A set of 6 contiguous techniques were taken from the individual attack sequence and the Viterbi algorithm (see below) was used to predict the tactics used for these six contiguous techniques (see also Step 5 illustrated above).
    - See also note on data volumes immediately following these bullets.
- Accuracy is presented by calculating 'total number of correct predictions of tactic/total number of technique observations validated' (as a percentage)
    - See also Results and Conclusion sections below where accuracy of results is presented.
    - The tests written also report on data failures so that the data may be more easily investigated for issues.
    - For all tests the number of technique observations safely exceeds 400 for each training set used.
    - All states (tactics) to be predicted are represented by example attack steps for each training set used.

Note on data volumes
Due to the average size of an attack sequence the training data was still greater than 400 observations per validation run after removal of a single attack. This provides enough data to build the HMM model to demonstrate here, however as described in the Results and Conclusion

sections as the data content increases (both in terms of volume and data content) the efficacy of the Viterbi algorithm will improve.

There does not seem to be specific advice on suitable data sizes for HMM. However, a general rule of thumb when sampling against distributions is 100 observations and more. The volume of data is safely beyond that limit. It also contains examples of transitions for all expected states. The discussion below illustrates some of the issues encountered when training sets have specific State to State transitions or State/Technique pairing missing. These are worked through and illustrated.

Two versions of this approach were developed (producing the same results). One used the data persisted as a graph database (see Loading the New Attack Model into an Example Database). The second persisted the data as simple python 'dicts'.

The results shown here (apart from the initial test) were from the 'dict' based version. This was simply because that this version became easier to maintain for this purpose.

Three sets of attack sequence examples (each progressively adding more data) are taken as input data and the results described below. As explained in the results and conclusions, this demonstrates some of the sensitivity inherent in the Viterbi approach to the data content.

7.3.2.3 Creating the HMM Matrices

This section provides more detail on how the HMM model was created for the test data (see also Markov Models).

When moving from one step to another in an attack sequence we have an example of a transition from one technique or tactic to an another (tactics TA0043 and TA0042 are not in scope see An Analysis Against ATT&CK Tactics/Techniques).

The transition probability matrix **A** was built to record transition probabilities from $StateTactic_{n1}$ to $StateTactic_{n2}$. Each a_{ij} in A represents the probability of transitioning from state *i* (row) to state *j* (column). This is done by walking through all the steps in the attack sequences, recording counts of transition from one tactic to another and finally calculating probabilities for a_{ij} by dividing each column in row_i by the total in row_i.

A simplified illustration is provided here. This is for the example set of transitions between State 1 (S_1) and State 2 (S_2) shown here {S_1-> S_1,S_1-> S_1,S_1-> S_2,S_2-> S_2,S_2-> S_2,S_2-> S_1}. For instance, here we can see 3 examples of S_1 transitioning to the next state. Out of these 3, S_1 transitions to S_1 2 times. So, in this test data we can say that if we are in state S_1 there is a 2/3 chance that we will transition to S_1.

	S_1	S_2	
S_1	2/3	1/3	Row ∑ =1
S_2	2/3	1/3	Row ∑ =1

Table 16 - Example State Transition Matrix Calculation

The emission/observation probability matrix **B** was built to record the probability of observing $Technique_m$ when in $State_n$. Each b_{ij} represents the probability $q_i(o_j)$ that is the probability of observing symbol o_j when in state q_i ($i \in [1,N]$, $j \in [1,M]$). Here N=number of different states/tactics, M=number of different observations/techniques. This is done by walking through all the steps in the attack sequences, recording counts of techniques observed (o) when using each tactic (state q) and finally calculating probabilities for b_{ij} = Total o_j / Total q_i.

A simplified illustration is provided here. This for the example set of tactic/techniques pairs (a pair is present in each step), = {(S_1,T_1),(S_1,T_1),(S_1,T_2),(S_1,T_3), (S_1,T_1), (S_2,T_2),(S_2,T_3),(S_2,T_1), (S_2,T_3), (S_2,T_2)}.

	T_1	T_2	T_3	
S_1	3/5	1/5	1/5	Row ∑ =1
S_2	1/5	2/5	1/5	Row ∑ =1

Table 17 - Example Observation/Emission Matrix Calculation

An initial probability distribution $\boldsymbol{\pi}$ (i.e. probability of each state being at the start of a sequence) was also built. This is based on the fact that all attacks will begin with the Tactic TA0001 Initial Access so this vector was given as the initial probability distribution [1,0,0,0,0,0,0,0,0,0,0,0]. Here the first element of this vector represents the probability of 1 for the starting tactic of an attack sequence will be TA0001 and that the probability of the starting tactic to be anything else is zero.

By hand building these parameters directly from the data in this way this represents what is termed the training stage of the model preparation.

The model was implemented in python code and can be found at (Maidens, 2023).

7.3.2.4 Leave One Out Approach

A 'leave one out cross validation' approach was then taken to create a simple measure of accuracy. In this approach the data is trained on all the attacks minus one attack. Then the

observation/technique sequence from the removed attack is used to predict (via the HMM model) the expected tactic sequence for the removed attack. This approach was repeated for every attack present in the set of attack sequences to provide an average success/error rate. A simple illustration for 3 attacks (the sets used here contain 26, 29 and 35 attacks respectively) is given below.

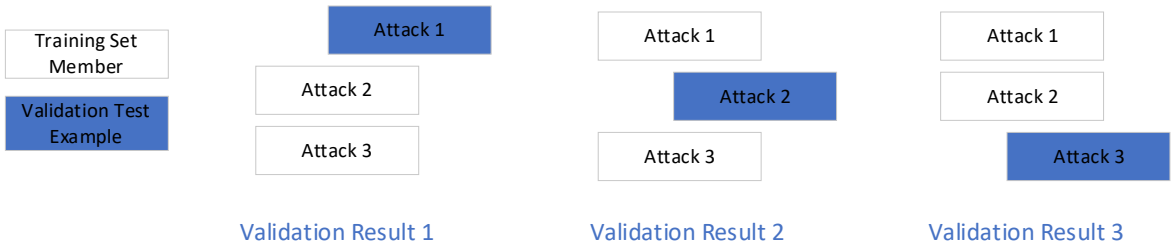


Figure 91 - Leave one out cross validation

7.3.2.5 The Viterbi Algorithm

The prediction was achieved using the Viterbi algorithm (the python code can be found at (Maidens, 2023)). In addition to the description given above (see Hidden Markov Models). A brief summary is also provided (for this standard algorithm) here (Kwok, 2019). I have found it easier to get a clearer view on the Viterbi algorithm through this recursive procedure (with description following) (the Results section below also provides some additional insight into how Viterbi behaves when unknown state transitions are encountered).

$\mu(X_0) = P[Y_0|X_0]P[X_0]$

$\mu(X_1) = \max(X_0) \quad \mu(X_0)\, P[X_1|X_0]P[Y_1|X_1]$

$\mu(X_2) = \max(X_1) \quad \mu(X_1)\, P[X_2|X_1]P[Y_2|X_2]$

……..

$\mu(X_n) = \max(X_{n-1}) \quad \mu(X_{n-1})\, P[X_n|X_{n-1}]P[Y_n|X_n]$

Here :

X_n is one of the hidden states (tactics)

Y_n is one of the observations (techniques observed)

The function P[A|B] can be read as the probability of A given (|) B.

This set of recursions is describing the algorithm moving from the starting observation (Y_0) and identifying the next estimated state through maximising the likelihood of observing that next state (using the probabilities in the Emission and State Transition matrices).

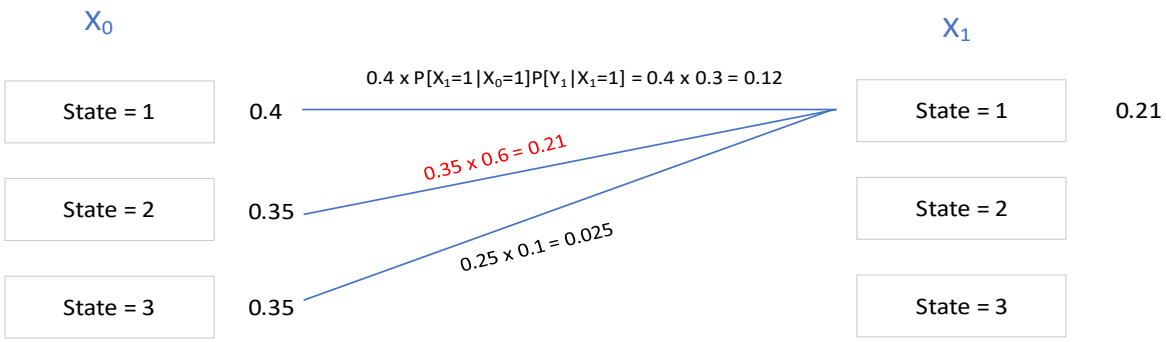
Also based on (Kwok, 2019) a toy example of calculating $\mu(X_0) = P[Y_0|X_0]P[X_0]$ and then the start of the next step. This is just to give an illustration of how the procedure unfolds and how the mostly likely values are calculated through the sequence.

$P[X_0]$ is derived from the initial probability distribution. $P[Y_0|X_0]$ from the B matrix

Step 0

X_0	Probability
State = 1	$P[Y_0\|X_0=1]P[X_0=1] = 0.4$
State = 2	0.35
State = 3	0.25

And the start for the next step $\mu(X_1) = \max(X_0) \quad \mu(X_0)\, P[X_1|X_0]P[Y_1|X_1]$ is



7.3.3 Results

7.3.3.1 Initial Test and Viterbi Failures

The initial test was run reading data persisted in a graph database.

The set of 26 attacks (a total of 397 observations providing an average training set of approximately 370 observations) used is given below

['admin@338_001', 'Lazarus_Group_001', 'Lazarus_Group_002', 'APT32_001', 'MuddyWater_001', 'MuddyWater_002', 'Mustang_Panda_001', 'Sandworm_001', 'Tropic_Trooper_001', 'APT28_001', 'APT28_002', 'APT28_003', 'APT28_004', 'APT29_001', 'APT29_002', 'APT29_003', 'APT29_004', 'APT41_001', 'APT41_002',

'menuPass_001', 'Carbanak_001', 'APT37_001', 'WizardSpider_001', 'OilRig_001', 'FIN7_001', 'APT3_001']

The results obtained are as follows:

26 attempts to predict the tactic sequence for the relevant validation observation sequence sample were run. 13 of the attempts to run the Viterbi algorithm and build the required 'hidden' sequence failed.

Of the tests that ran the accuracy achieved was 100% (that is all Tactics for the Techniques presented were correctly predicted).

7.3.3.1.1 Discussion

Due to the high number of failed validations the approach was investigated and refined.

The Viterbi algorithm originally used was from the standard python sklearn package (hmm.MultinomialHMM was used as this is categorical data).

The Viterbi algorithm was reimplemented as a function within the python code used to run the tests. This was done to provide full control over the algorithm and to add detailed tracing. As a result, the source of the problem became quickly apparent and is described below.

As described above the Viterbi algorithm works recursively through the observation sequence calculating the next most likely state. So, for step n we have:

$\mu(X_n) = \max(X_{n-1}) \quad \mu(X_{n-1})\, P[X_n|X_{n-1}]P[Y_n|X_n]$

Therefore, if a point in the sequence is reached where either $P[X_n|X_{n-1}]$ (probability of state to state transition) or $P[Y_n|X_n]$ (probability of observing Y while in state X) can only be 0 then no prediction can be realistically made as all path likelihoods will reduce to zero. In the case of the sklearn algorithm it looks like the algorithm tries to continue by selecting one state from all states with the same maximum likelihood (here it seems it would select from a list of states in the order provided to the algorithm coincidentally in this case it resolved to TA0040).

The reason why this point was being reached was when the example attack removed from the set of attacks to train contained a transition that was not present in the training set (i.e. was unique to that attack only).

Two steps were taken to resolve this issue:

- The approach coded was refined further

- When the validation extract is built the training set is analysed for transitions types that exist within it. Construction of the extract is terminated at the point where a non-existent transition is found as the Viterbi algorithm would report incorrect results from this point anyway (a record of this is kept in the test rig)
- When the results are reported the statistics include information about what length of observations were eventually validated (compared with the total originally intended). A record of the transitions that are unique in the validation extract are also recorded.

- More attacks were also documented to increase the range of data in the training sets

7.3.3.2 Second Test Using Improved Code and Increased Data

The second test was run reading data persisted in python dicts (this became more maintainable than the graph database example for the testing purposes).

The set of 29 attacks used is given below (including 437 observations – Average training set > 420):

['admin@338_001', 'Lazarus_Group_001', 'Lazarus_Group_002', 'APT32_001', 'MuddyWater_001', 'MuddyWater_002', 'Mustang_Panda_001', 'Sandworm_001', 'Tropic_Trooper_001', 'APT28_001', 'APT28_002', 'APT28_003', 'APT28_004', 'APT29_001', 'APT29_002', 'APT29_003', 'APT29_004', 'APT41_001', 'APT41_002', 'menuPass_001', 'Carbanak_001', 'APT37_001', 'WizardSpider_001', 'OilRig_001', 'FIN7_001', 'APT3_001', 'Ajax_Security_Team_001', 'Andariel_001', 'APT38_001']

The results obtained are as follows:

Final results for 29 validations :

FullPreds : 15

PartialPreds : 11

NoPreds : 3

TotalObsPresented : 166

TotalObsPredicted : 125

TotalInvalidPairsPresented : 25

TotalSuccessfulPredictions : 125

TotalFailedPredictions : 0

This means that :

Out of the 29 attacks validated, 15 were now run successfully against the training set (that is with a complete observation fragment) – 51.7% were fully validated.

11 attacks were still partially validated (that is a portion of the fragment could be predicted) - 89.7% were fully or partially validated.

3 attacks had no predictions made at all – 10.3% could not be validated.

25 different pairs (Tactic to Tactic transition or Technique associated with a Tactic) were found over the course of all the tests that were not repeated in the training set. These are reported by the test rig. A sample of output is shown here (but for brevity none shown further).

> As noted in the results 14 validations were found to have some sort of problem, therefore there would in fact be a total of 14 diagnostic outputs here.
>
> For each problem found this allows the source Attack to be understood and detailed results. The results include a list of unique 'TacticTech' or 'TacticTactic' pairs. This allows further discovery about the nature of the data attack sequences.

{'**AttackName**': 'MuddyWater_002', '**ObsExt**': ['T1566.001', 'T1204.002', 'T1547.001', 'T1218.003', 'T1140', 'T1059.001'], '**StateExt**': ['TA0001', 'TA0002', 'TA0003', 'TA0005', 'TA0005', 'TA0002'], '**PredState**': ['TA0001', 'TA0002', 'TA0003'], '**IntendedObsLen**': 6, '**ActualObsLen**': 3, '**InvalidPairs**': [{'Type': 'TacticTech', 'Tactic': 'TA0005', 'Tech': 'T1218.003'}], '**CorrectPred**': 3, '**IncorrectPred**': 0}

{'**AttackName**': 'Sandworm_001', '**ObsExt**': ['T1566.001', 'T1204.002', 'T1203', 'T1105', 'T1547', 'T1056.001'], '**StateExt**': ['TA0001', 'TA0002', 'TA0002', 'TA0011', 'TA0003', 'TA0006'], '**PredState**': ['TA0001', 'TA0002', 'TA0002', 'TA0011'], '**IntendedObsLen**': 6, '**ActualObsLen**': 4, '**InvalidPairs**': [{'Type': 'TacticTech', 'Tactic': 'TA0003', 'Tech': 'T1547'}, {'Type': 'TacticTech', 'Tactic': 'TA0006', 'Tech': 'T1056.001'}, {'Type': 'TacticTactic', 'PrevTactic': 'TA0003', 'Tactic': 'TA0006'}], '**CorrectPred**': 4, '**IncorrectPred**': 0}

{'**AttackName**': 'APT29_004', 'ObsExt': ['T1021.002', 'T1570', 'T1213'], '**StateExt**': ['TA0008', 'TA0008', 'TA0009'], '**PredState**': [], '**IntendedObsLen**': 3, '**ActualObsLen**': 0, '**InvalidPairs**': [{'Type': 'TacticTech', 'Tactic': 'TA0008', 'Tech': 'T1021.002'}, {'Type': 'TacticTactic', 'PrevTactic': 'TA0008', 'Tactic': 'TA0008'}, {'Type': 'TacticTactic', 'PrevTactic': 'TA0008', 'Tactic': 'TA0009'}], '**CorrectPred**': 0, '**IncorrectPred**': 0}

Out of a possible total of 166 test predictions, 125 predictions could be made – 75.3% of all possible predictions could be made.

Out of the 125 predictions 125 were successful (that is 100% accuracy)

7.3.3.3 Final Test Using Extended Set of Example Attacks

The third test was run again reading data persisted in python dicts.

The set of 35 attacks used is given below (including 537 observations – Average training set > 420):

['admin@338_001', 'Lazarus_Group_001', 'Lazarus_Group_002', 'APT32_001', 'MuddyWater_001', 'MuddyWater_002', 'Mustang_Panda_001', 'Sandworm_001', 'Tropic_Trooper_001', 'APT28_001', 'APT28_002', 'APT28_003', 'APT28_004', 'APT29_001', 'APT29_002', 'APT29_003', 'APT29_004', 'APT41_001', 'APT41_002',

'menuPass_001', 'Carbanak_001', 'APT37_001', 'WizardSpider_001', 'OilRig_001', 'FIN7_001', 'APT3_001', 'Ajax_Security_Team_001', 'Andariel_001', 'APT38_001', 'ZAPT33_001', 'ZAPT19_001', 'ZSandworm_002', 'ZAPT28_005', 'ZAPT32_001', 'ZAPT29_005']

The results obtained are as follows:

Final results for 35 validations :

FullPreds : 25

PartialPreds : 7

NoPreds : 3

TotalObsPresented : 202

TotalObsPredicted : 169

TotalInvalidPairsPresented : 17

TotalSuccessfulPredictions : 168

TotalFailedPredictions : 1

This means that:

Out of the 35 attacks validated, 25 were now run successfully against the training set (that is with a complete observation fragment) – 71.4% were fully validated.

7 attacks were still partially validated (that is a portion of the fragment could be predicted) - 91.4% were fully or partially validated.

3 attacks had no predictions made at all – 8.6% could not be validated.

17 different pairs (Tactic to Tactic transition or Technique associated with a Tactic) were found over the course of all the tests that were not repeated in the training set. These are reported by the test rig. A sample of output is shown here (but for brevity none shown further).

Out of a possible total of 202 test predictions, 169 predictions could be made – 75.3% of all possible predictions could be made

Out of the 169 predictions 168 were successful (still virtually 100% accuracy)

As noted above, the code reports on failed predictions. The following attacks had actual incorrect predictions in them

{'**AttackName**': 'ZSandworm_002',
'**ObsExt**': ['T1566.001', 'T1204.002', 'T1203', 'T1105', 'T1547', 'T1547.001'],
'**StateExt**': ['TA0001', 'TA0002', 'TA0002', 'TA0011', 'TA0003', 'TA0004'],
'**PredState**': ['TA0001', 'TA0002', 'TA0002', 'TA0011', 'TA0003', 'TA0003'],
'**IntendedObsLen**': 6, '**ActualObsLen**': 6, '**InvalidPairs**': [], '**CorrectPred**': 5, '**IncorrectPred**': 1}

This says that for ZSandworm_002 the Observation sequence extracted was ['T1566.001', 'T1204.002', 'T1203', 'T1105', 'T1547', 'T1547.001'] the expected State sequence was ['TA0001', 'TA0002', 'TA0002', 'TA0011', 'TA0003', 'TA0004'] the predicted sequence was wrong at the last step where TA0003 was predicted instead of the expected TA0004.

T1547.001 is Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder and is associated with Tactic TA0003 Persistence and TA0004 Privilege Escalation.

Within test data there are 16 examples of T1547.001 associated with TA0003 and only 2 examples associated with TA0004. This explains this prediction. It is also example of where this result could be of interest to analysts in identifying anomalous behaviours.

7.3.4 Conclusion

The test runs have been constructed to demonstrate usage of the proposed data model as the data changes.

The table below provides a summary of the results shown above

	Attacks Run	Attacks Run or Partially Run	Attacks – No Predictions	Tactic Predictions Made	Failed Predictions	Accuracy of Predictions Made	Accuracy – Predictions Intended	Invalid Pairs
Test 1	50%	50%	50%	N/A	N/A	100%	NA	N/A
Test 2	51.7%	89.7%	10.3%	75.3%	0	100%	75.3%	25
Test 3	71.4%	91.4%	8.6%	83.6%	1	99.4%	83.2%	17

Accuracy of predictions actually made seems very high. This is perhaps a reflection of the high number of techniques that are only related to a single Tactic.

Although this should only be treated as illustrative there is a trend towards increased capability to make predictions as more data is introduced. This should be expected as naturally more combinations will be introduced with a larger sample of real-world attack sequences.

The code produced diagnostics that could have been used to select and record sequences of attacks that would provide missing Tactic/Tactic or Tactic/Sequence combinations. This could have been used to create an ‘internally consistent’ sample of attacks. This temptation was resisted as the results obtained here provide a better insight into how a real word database of attack sequences may behave.

There are a number of different reasons why failed predictions may occur

- The attack itself is unusual and the failed prediction is actually an alert to consider more detailed analysis of the behaviours. This may be a signal of a new actor or new behaviours adopted
- More simply it may be a case of a number of options having the same probability and the Viterbi algorithm simply picking one of the matching options.
- It could also be that the data collected does not match real world distributions of statistics relating to Tactic/Tactic transition or Tactic/Technique pairings. A real database of such examples (for instance providing open access for input to relate research themes) would benefit from regular review of the data statistics and comparison with such things as annual summary reports (e.g. (ENISA, 2021)).

7.4 Using the Attack Model – Markov Model

7.4.1 Introduction

This is the next example as outlined in Introduction.

The attack sequences stored in the model described above (A New Attack Model) can be used to understand the transitions between various Tactic and Techniques described in the attacks.

A common assumption in research is that cyber-attacks may be considered as Markov processes (see also Markov Models) (e.g. (Dass et al., 2021), (Ye et al., 2004),(Chadza et al., 2020)).

This section uses the technique sequences in the attacks and builds a Markov model (MM) to predict the next technique to be used. This useful because we can use this information to try and predict 'next steps' of an attacker given an observed technique stream. For this example, Recurrent Neural Networks or Long Short-Term Memory approaches could also be used but here a simpler Markov Model is used for illustration.

In the section below:

- The approach is described in Approach
- The results of three tests are described in Results
- A summary of conclusions is discussed in Conclusion

7.4.2 Approach

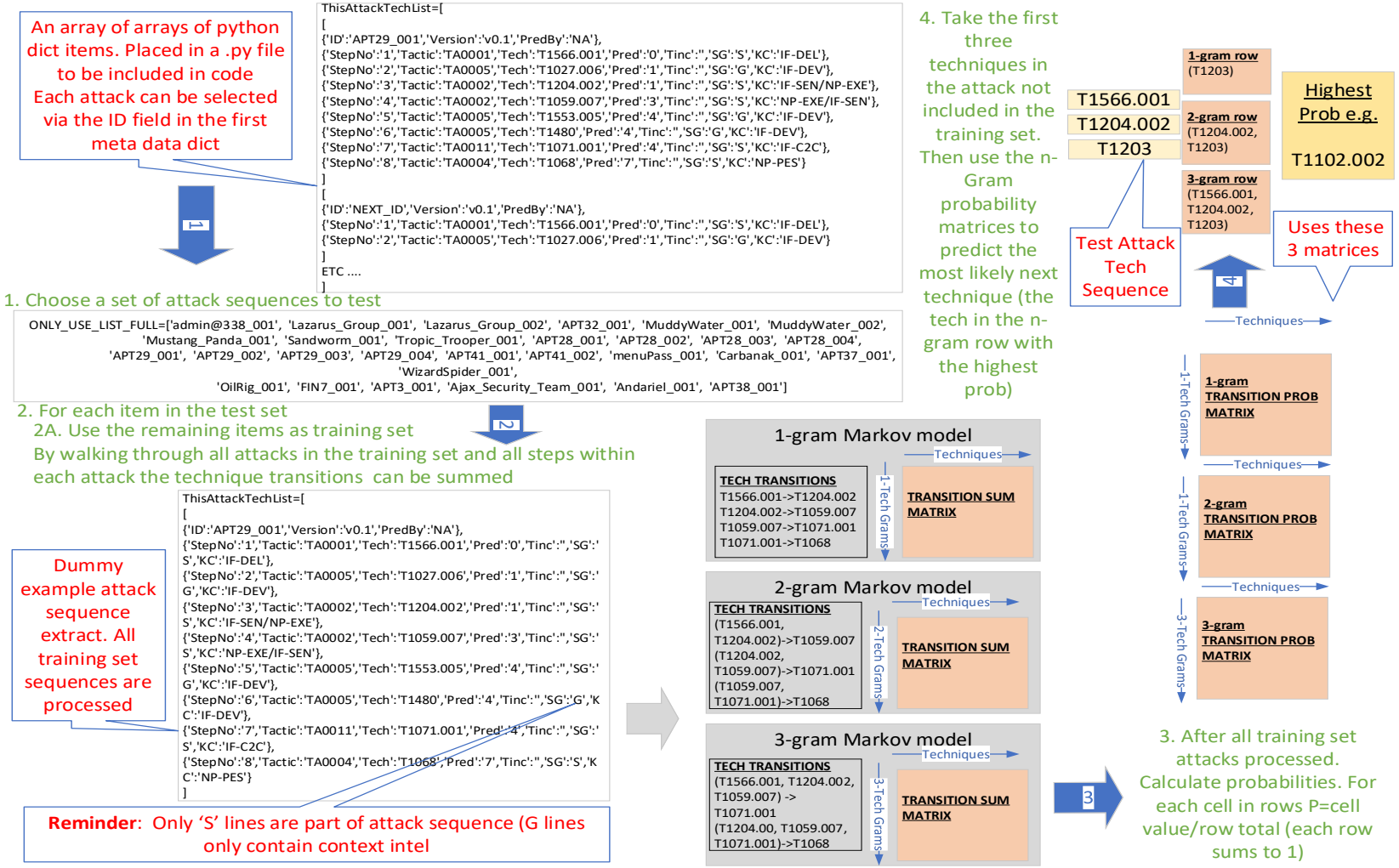


Figure 92 - Building n-gram Markov model matrices

A summary of the approach is provided here with supporting detail below:

- Code and data used for this test may be found at (Maidens, 2023).
    - See also Data Preparation that describes how the data was prepared for this demonstration
- The set of attack sequence examples is taken as input data.
    - Two versions of this approach were developed (producing the same results). One used the data persisted as a graph database (see Loading the New Attack Model into an Example Database). The second persisted the data as simple python 'dicts.
    - The results shown here were from the 'dict' based version. This was simply because that this version became easier to maintain for this purpose.
- For **each individual attack** in the example set (this is called a Leave One Out approach described also in Leave One Out Approach)
    - The individual attack was taken out of the whole set.
    - The rest of the example set was used as the training set
    - The training set was used to create MM transition probability matrices (as described below)
        - See also Steps 2 and 3 in the diagram above
    - A set of 3 contiguous techniques were taken from the individual attack sequence then either 1, 2 or 3 of these techniques (1-gram, 2-gram, 3-gram) were then used to predict the next likely technique (see also MM Transition Probability Matrix and n-gram Models below).
        - See also Step 4 in the diagram above.
- Due to the average size of an attack sequence the training data was still greater than 400 observations per validation run after removal of a single attack. This provides enough data to build the MM model to demonstrate here.
- Various accuracy measures were calculated and recorded below.
    - The tests written also report on data failures so that the data may be more easily investigated for issues.
    - These are discussed fully in Results section below.

7.4.2.1 MM Transition Probability Matrix and n-gram Models

The transition probability matrix **A** was built to record transition probabilities from $StateTechnique_{n1}$ to $StateTechnique_{n2}$. Each a_{ij} in A represents the probability of transitioning from state *i* to state *j* . This is done by walking through all the steps in the attack sequences, recording

counts of transition from one technique to another and finally calculating probabilities for a_{ij} = Total n_i to n_j / Total n_i.

A simplified illustration is provided here. This for the example set of transitions between Technique 1 (T_1) and Technique 2 (T_2) = {T_1-> T_1,T_1-> T_1,T_1-> T_2,T_2-> T_2,T_2-> T_2,T_2-> T_1}.

	T_1	**T_2**	
T_1	2/3	1/3	Row ∑ =1
T_2	2/3	1/3	Row ∑ =1

Table 18 - Example Technique Transition Matrix Calculation

By hand building these parameters directly from the data this represents the training stage of the model preparation.

For building n-Markov (**that is n-gram**) models the approach is essentially the same (see also illustration above).

For n>1 the rows are allocated to tuples instead of single techniques.

For instance, for a set of examples for techniques T_1, T_2, T_3, T_4 that includes the following sequences

{T_1, T_2, T_1, T_3}, {T_2, T_1, T_3, T_4}, {T_2, T_3, T_4, T_1} ……

If we were predicting the probabilities of the fourth technique in the sequence using a 2-Markov model, rows in the Markov matrix would include those for tuples {T_2, T_1}, {T_1, T_3}, {T_3, T_4} with appropriate probabilities for T_1, T_2, T_3, T_4 in the columns.

In the examples below 1, 2 and 3 Markov models are implemented.

7.4.3 Results

This section provides a summary of the results obtained for the 1, 2 and 3-gram Markov Models (see also MM Transition Probability Matrix and n-gram Models). These are then discussed in the following Conclusions section.

The first result output also provides a summary of the information in the output.

7.4.3.1 1-gram Markov model

TEST_SET_ORIGINAL=['admin@338_001', 'Lazarus_Group_001', 'Lazarus_Group_002', 'APT32_001', 'MuddyWater_001', 'MuddyWater_002', 'Mustang_Panda_001', 'Sandworm_001', 'Tropic_Trooper_001', 'APT28_001', 'APT28_002', 'APT28_003', 'APT28_004', 'APT29_001', 'APT29_002', 'APT29_003', 'APT29_004', 'APT41_001', 'APT41_002', 'menuPass_001', 'Carbanak_001', 'APT37_001', 'WizardSpider_001', 'OilRig_001', 'FIN7_001', 'APT3_001']

Total tests is 26

Final result (1) is success=6/failed=17/unable=1/too short=2

Overall accuracy % is 23.076923076923077

Accuracy % against runnable is 26.08695652173913

Fail list is [{'Attack': 'APT28_002', 'Depth Tuple': ['T1114']}]

Short fail list is ['APT29_004', 'APT41_001']

The meaning of the output for this example is explained here. This structure of this explanation is valid for the following result outputs.

TEST_SET_*=[….] provides a record of the attack sequences used in this test.

The output above notes (in **Total Tests**) that 26 tests were run (the number of attacks in the test set).

It also summarises which tests were successful (in **Final Result** – the (1) indicates the length of the n-gram used). Here:

6 tests were run where the next expected technique was successfully predicted

17 tests were run where the next expected technique was not successfully predicted

1 test could not be run at all

For failed tests (given in '**Fail list is'**) the attack name and the Tuple used to predict is displayed

Here APT28_002 failed and it was using T1114 to predict using the MM A matrix

In this case it appears that this attack was the only one that used the technique T1114 so when it was removed (during the leave one out validation process) this means it would no longer be in the training set.

2 tests were too short.

This means that given the observation length being sampled (3 in this case) there were not enough steps in these attacks to both sample and have a follow-on technique to predict. Both APT29_004 and APT41_001 (shown in **Short fail list**) only have three steps.

The overall accuracy percentage is reported in this case 6 out of 26

The runnable accuracy percentage is reported in this case 6 out of 23 (26-(1+2))

TEST_SET_NEXT =['admin@338_001', 'Lazarus_Group_001', 'Lazarus_Group_002', 'APT32_001', 'MuddyWater_001', 'MuddyWater_002', 'Mustang_Panda_001', 'Sandworm_001', 'Tropic_Trooper_001', 'APT28_001', 'APT28_002', 'APT28_003', 'APT28_004', 'APT29_001', 'APT29_002', 'APT29_003', 'APT29_004', 'APT41_001', 'APT41_002', 'menuPass_001', 'Carbanak_001', 'APT37_001', 'WizardSpider_001', 'OilRig_001', 'FIN7_001', 'APT3_001', 'Ajax_Security_Team_001', 'Andariel_001', 'APT38_001']

Total tests is 29

Final result (1) is success=7/failed=19/unable=1/too short=2

Overall accuracy % is 24.137931034482758

Accuracy % against runnable is 26.923076923076923

Fail list is [{'Attack': 'APT28_002', 'Depth Tuple': ['T1114']}]

Short fail list is ['APT29_004', 'APT41_001']

TEST_SET_FULL ['admin@338_001', 'Lazarus_Group_001', 'Lazarus_Group_002', 'APT32_001', 'MuddyWater_001', 'MuddyWater_002', 'Mustang_Panda_001', 'Sandworm_001', 'Tropic_Trooper_001', 'APT28_001', 'APT28_002', 'APT28_003', 'APT28_004', 'APT29_001', 'APT29_002', 'APT29_003', 'APT29_004', 'APT41_001', 'APT41_002', 'menuPass_001', 'Carbanak_001', 'APT37_001', 'WizardSpider_001', 'OilRig_001', 'FIN7_001', 'APT3_001', 'Ajax_Security_Team_001', 'Andariel_001', 'APT38_001', 'ZAPT33_001', 'ZAPT19_001', 'ZSandworm_002', 'ZAPT28_005', 'ZAPT32_001', 'ZAPT29_005']

Total tests is 35

Final result (1) is success=8/failed=24/unable=1/too short=2

Overall accuracy % is 22.857142857142858

Accuracy % against runnable is 25.0

Fail list is [{'Attack': 'APT28_002', 'Depth Tuple': ['T1114']}]

Short fail list is ['APT29_004', 'APT41_001']

7.4.3.2 2-gram Markov model

TEST_SET_ORIGINAL

Total tests is 26

Final result (2) is success=4/failed=9/unable=11/too short=2

Overall accuracy % is 15.384615384615385

Accuracy % against runnable is 30.76923076923077

Fail list is [{'Attack': 'APT32_001', 'Depth Tuple': ['T1204.002', 'T1053.005']}, {'Attack': 'Mustang_Panda_001', 'Depth Tuple': ['T1204.002', 'T1574.002']}, {'Attack': 'APT28_001', 'Depth Tuple': ['T1190', 'T1105']}, {'Attack': 'APT28_002', 'Depth Tuple': ['T1078', 'T1114']}, {'Attack': 'APT28_003', 'Depth Tuple': ['T1078', 'T1105']}, {'Attack': 'APT28_004', 'Depth Tuple': ['T1570', 'T1071.001']}, {'Attack': 'APT29_001', 'Depth Tuple': ['T1204.002', 'T1059.007']}, {'Attack': 'APT29_002', 'Depth Tuple': ['T1204.002', 'T1071.001']}, {'Attack': 'APT37_001', 'Depth Tuple': ['T1189', 'T1204.001']}, {'Attack': 'FIN7_001', 'Depth Tuple': ['T1204.002', 'T1218.005']}, {'Attack': 'APT3_001', 'Depth Tuple': ['T1204.001', 'T1189']}]

Short fail list is ['APT29_004', 'APT41_001']

TEST_SET_NEXT

Total tests is 29

Final result (2) is success=4/failed=9/unable=14/too short=2

Overall accuracy % is 13.793103448275861

Accuracy % against runnable is 30.76923076923077

Fail list is [{'Attack': 'APT32_001', 'Depth Tuple': ['T1204.002', 'T1053.005']}, {'Attack': 'Mustang_Panda_001', 'Depth Tuple': ['T1204.002', 'T1574.002']}, {'Attack': 'APT28_001', 'Depth Tuple': ['T1190', 'T1105']}, {'Attack': 'APT28_002', 'Depth Tuple': ['T1078', 'T1114']}, {'Attack': 'APT28_003', 'Depth Tuple': ['T1078', 'T1105']}, {'Attack': 'APT28_004', 'Depth Tuple': ['T1570', 'T1071.001']}, {'Attack': 'APT29_001', 'Depth Tuple': ['T1204.002', 'T1059.007']}, {'Attack': 'APT29_002', 'Depth Tuple': ['T1204.002', 'T1071.001']}, {'Attack': 'APT37_001', 'Depth Tuple': ['T1189', 'T1204.001']}, {'Attack': 'FIN7_001', 'Depth Tuple': ['T1204.002', 'T1218.005']}, {'Attack': 'APT3_001', 'Depth Tuple': ['T1204.001', 'T1189']}, {'Attack': 'Ajax_Security_Team_001', 'Depth Tuple': ['T1204.002', 'T1059']}, {'Attack': 'Andariel_001', 'Depth Tuple': ['T1204.002', 'T1140']}, {'Attack': 'APT38_001', 'Depth Tuple': ['T1204.001', 'T1059.001']}]

Short fail list is ['APT29_004', 'APT41_001']

TEST_SET_FULL

Total tests is 35

Final result (2) is success=9/failed=13/unable=11/too short=2

Overall accuracy % is 25.71428571428571

Accuracy % against runnable is 40.909090909090914

Fail list is [{'Attack': 'APT28_001', 'Depth Tuple': ['T1190', 'T1105']}, {'Attack': 'APT28_002', 'Depth Tuple': ['T1078', 'T1114']}, {'Attack': 'APT28_003', 'Depth Tuple': ['T1078', 'T1105']}, {'Attack': 'APT28_004', 'Depth Tuple': ['T1570', 'T1071.001']}, {'Attack': 'APT37_001', 'Depth Tuple': ['T1189', 'T1204.001']}, {'Attack': 'FIN7_001', 'Depth Tuple': ['T1204.002', 'T1218.005']}, {'Attack': 'APT3_001', 'Depth Tuple': ['T1204.001', 'T1189']}, {'Attack': 'Ajax_Security_Team_001', 'Depth Tuple': ['T1204.002', 'T1059']}, {'Attack': 'Andariel_001', 'Depth Tuple': ['T1204.002', 'T1140']}, {'Attack': 'APT38_001', 'Depth Tuple': ['T1204.001', 'T1059.001']}, {'Attack': 'ZAPT33_001', 'Depth Tuple': ['T1204.001', 'T1105']}]

Short fail list is ['APT29_004', 'APT41_001']

7.4.3.3 3-gram Markov model

TEST_SET_ORIGINAL

Total tests is 26

Final result (3) is success=3/failed=6/unable=15/too short=2

Overall accuracy % is 11.538461538461538

Accuracy % against runnable is 33.33333333333333

Fail list is [{'Attack': 'APT32_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1053.005']}, {'Attack': 'Mustang_Panda_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1574.002']}, {'Attack': 'APT28_001', 'Depth Tuple': ['T1078', 'T1190', 'T1105']}, {'Attack': 'APT28_002', 'Depth Tuple': ['T1566.002', 'T1078', 'T1114']}, {'Attack': 'APT28_003', 'Depth Tuple': ['T1566.002', 'T1078', 'T1105']}, {'Attack': 'APT28_004', 'Depth Tuple': ['T1078', 'T1570', 'T1071.001']}, {'Attack': 'APT29_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059.007']}, {'Attack': 'APT29_002', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1071.001']}, {'Attack': 'APT41_002', 'Depth Tuple': ['T1195.002', 'T1140', 'T1059.003']}, {'Attack': 'menuPass_001', 'Depth Tuple': ['T1190', 'T1574.002', 'T1140']}, {'Attack': 'APT37_001', 'Depth Tuple': ['T1566.002', 'T1189', 'T1204.001']}, {'Attack': 'WizardSpider_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059.001']}, {'Attack': 'OilRig_001', 'Depth Tuple': ['T1566.003', 'T1059.005', 'T1140']}, {'Attack': 'FIN7_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1218.005']}, {'Attack': 'APT3_001', 'Depth Tuple': ['T1566.002', 'T1204.001', 'T1189']}]

Short fail list is ['APT29_004', 'APT41_001']

TEST_SET_NEXT

Total tests is 29

Final result (3) is success=3/failed=6/unable=18/too short=2

Overall accuracy % is 10.344827586206897

Accuracy % against runnable is 33.33333333333333

Fail list is [{'Attack': 'APT32_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1053.005']}, {'Attack': 'Mustang_Panda_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1574.002']}, {'Attack': 'APT28_001', 'Depth Tuple': ['T1078', 'T1190', 'T1105']}, {'Attack': 'APT28_002', 'Depth Tuple': ['T1566.002', 'T1078', 'T1114']}, {'Attack': 'APT28_003', 'Depth Tuple': ['T1566.002', 'T1078', 'T1105']}, {'Attack': 'APT28_004', 'Depth Tuple': ['T1078', 'T1570', 'T1071.001']}, {'Attack': 'APT29_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059.007']}, {'Attack': 'APT29_002', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1071.001']}, {'Attack': 'APT41_002', 'Depth Tuple': ['T1195.002', 'T1140', 'T1059.003']}, {'Attack': 'menuPass_001', 'Depth Tuple': ['T1190', 'T1574.002', 'T1140']}, {'Attack': 'APT37_001', 'Depth Tuple': ['T1566.002', 'T1189', 'T1204.001']}, {'Attack': 'WizardSpider_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059.001']}, {'Attack': 'OilRig_001', 'Depth Tuple': ['T1566.003', 'T1059.005', 'T1140']}, {'Attack': 'FIN7_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1218.005']}, {'Attack': 'APT3_001', 'Depth Tuple': ['T1566.002', 'T1204.001', 'T1189']}, {'Attack': 'Ajax_Security_Team_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059']}, {'Attack': 'Andariel_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1140']}, {'Attack': 'APT38_001', 'Depth Tuple': ['T1566.002', 'T1204.001', 'T1059.001']}]

Short fail list is ['APT29_004', 'APT41_001']

TEST_SET_FULL

Total tests is 35

Final result (3) is success=8/failed=10/unable=15/too short=2

Overall accuracy % is 22.857142857142858

Accuracy % against runnable is 44.44444444444444

Fail list is [{'Attack': 'APT28_001', 'Depth Tuple': ['T1078', 'T1190', 'T1105']}, {'Attack': 'APT28_002', 'Depth Tuple': ['T1566.002', 'T1078', 'T1114']}, {'Attack': 'APT28_003', 'Depth Tuple': ['T1566.002', 'T1078', 'T1105']}, {'Attack': 'APT28_004', 'Depth Tuple': ['T1078', 'T1570', 'T1071.001']}, {'Attack': 'APT41_002', 'Depth Tuple': ['T1195.002', 'T1140', 'T1059.003']}, {'Attack': 'menuPass_001', 'Depth Tuple': ['T1190', 'T1574.002', 'T1140']}, {'Attack': 'APT37_001', 'Depth Tuple': ['T1566.002', 'T1189', 'T1204.001']}, {'Attack': 'WizardSpider_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059.001']}, {'Attack': 'OilRig_001', 'Depth Tuple': ['T1566.003', 'T1059.005', 'T1140']}, {'Attack': 'FIN7_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1218.005']}, {'Attack': 'APT3_001', 'Depth Tuple': ['T1566.002', 'T1204.001', 'T1189']}, {'Attack': 'Ajax_Security_Team_001',

'Depth Tuple': ['T1566.001', 'T1204.002', 'T1059']}, {'Attack': 'Andariel_001', 'Depth Tuple': ['T1566.001', 'T1204.002', 'T1140']}, {'Attack': 'APT38_001', 'Depth Tuple': ['T1566.002', 'T1204.001', 'T1059.001']}, {'Attack': 'ZAPT33_001', 'Depth Tuple': ['T1566.002', 'T1204.001', 'T1105']}]

Short fail list is ['APT29_004', 'APT41_001']

7.4.4 Conclusion

Derived from the results above here is a summary of the key observations

Accuracy percentages

		ORIGINAL (26)	NEXT (29)	FULL (35)
1-gram Markov	*Overall*	23.1	24.1	22.9
	Runnable	26.1	26.9	25.0
2-gram Markov	*Overall*	15.4	13.8	25.7
	Runnable	30.8	30.8	40.9
3-gram Markov	*Overall*	11.5	10.3	22.9
	Runnable	33.3	33.3	44.4

Failure percentages and counts

		ORIGINAL (26)	NEXT (29)	FULL (35)
1-gram Markov	*Failed*	73.9	73.1	75
	Unable	1/26*100	1/29*100	1/35*100
2-gram Markov	*Failed*	69.2	69.2	59.1
	Unable	11/26*100	14/29*100	11/35*100
3-gram Markov	*Failed*	66.7	66.7	55.6
	Unable	15/25*100	18/29*100	15/29*100

The overall prediction success from these experiments was quite low. However, the results were still reported as they provide insight in their own terms.

In general, we can see the accuracy of runnable tests rising as the data size and n-gram level is increased.

However, as the n-gram level increases we can also see a tendency to increase the number of predictions that cannot be run. This happens when the validation set used to train the model does not include the specific n-gram tuple in the sequence being tested. As n increases then the number of possible combinations included in tuples also increases. This effectively increases the chance of any one test sequence containing a unique n-gram.

The best overall accuracy of 25.7% was achieved with the largest set of attacks but using a 2-gram Markov prediction.

The best runnable accuracy of 44.4% was achieved with the largest set of attacks but using a 3-gram Markov prediction.

It is reasonable to expect that in a real-world database the failure rates would be lower due to the wider range of data combinations. Missing tuples may also be informative in their own right. This may indicate anomalous behaviours that should be analysed.

As noted with the HMM model, the code produced diagnostics that could have been used to select and record sequences of attacks that would provide missing Technique tuples. This could have been used to create an 'internally consistent' sample of attacks. This temptation was resisted as the results obtained here provide a better insight into how a real word database of attack sequences may behave. For instance, in this data set the tuples identified as missing have a number of repeated pair combinations. A number of these are related to techniques T1204.001 and T1204.002 with following techniques.

As also noted with the HMM model, a real database of such examples (for instance providing open access for input to relate research themes) would benefit from regular review of the data integrity and statistics including comparison with such things as annual summary reports (e.g. (ENISA, 2021)).

7.5 Using the Attack Model – Unified Kill Chain

7.5.1 Introduction

This is the next example as outlined in Introduction.

The ATT&CK tactic model makes no assumptions about sequence.

By using the records of the attack sequences this section hopes to explore the relationship between the Tactics and an implicit high-level sequence provided by a Kill Chain model.

In the section below:

- The approach is described in Approach
- The results and conclusions are combined in Results and Conclusions

7.5.2 Approach

The approach here is quite straightforward and is intended primarily as an illustration of additional intelligence provided by the structuring of the attack sequences.

The attack sequences were analysed and the combinations of ATT&CK Tactic and UKC Steps in the various steps of the sequences were recorded.

The code used for this may be found at (Maidens, 2023).

In this case the results immediately provide illustrations of the conclusions so the sections have been combined.

7.5.3 Results and Conclusions

The results below provide a frequency count of mappings between the UKC-Chain Steps and the ATT&CK Tactics is given below. A Heat Map representation is also provided to give a simpler visual view.

	TA0043	TA0042	TA0001	TA0002	TA0003	TA0004	TA0005	TA0006	TA0007	TA0008	TA0009	TA0011	TA0010	TA0040
IF-REC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IF-WEP	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IF-DEL	0.0	0.0	24.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IF-SEN	0.0	0.0	19.0	17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IF-EXP	0.0	0.0	8.0	5.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IF-PER	0.0	0.0	0.0	1.0	22.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
IF-DEV	0.0	0.0	0.0	0.0	0.0	1.0	87.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
IF-C2C	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	135.0	5.0	0.0
NP-DIS	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	52.0	0.0	1.0	0.0	0.0	0.0
NP-PES	0.0	0.0	0.0	0.0	0.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NP-EXE	0.0	0.0	0.0	59.0	2.0	1.0	30.0	1.0	1.0	0.0	2.0	3.0	0.0	0.0
NP-CAC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.0	0.0	0.0	0.0	0.0	0.0	0.0
NP-LMV	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	6.0	0.0	0.0	0.0	0.0
AO-COL	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	29.0	0.0	0.0	0.0
AO-EXF	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	20.0	0.0
AO-TMA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0
AO-OBJ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 93 - UKC Step to ATT&CK Tactic Frequency Mapping

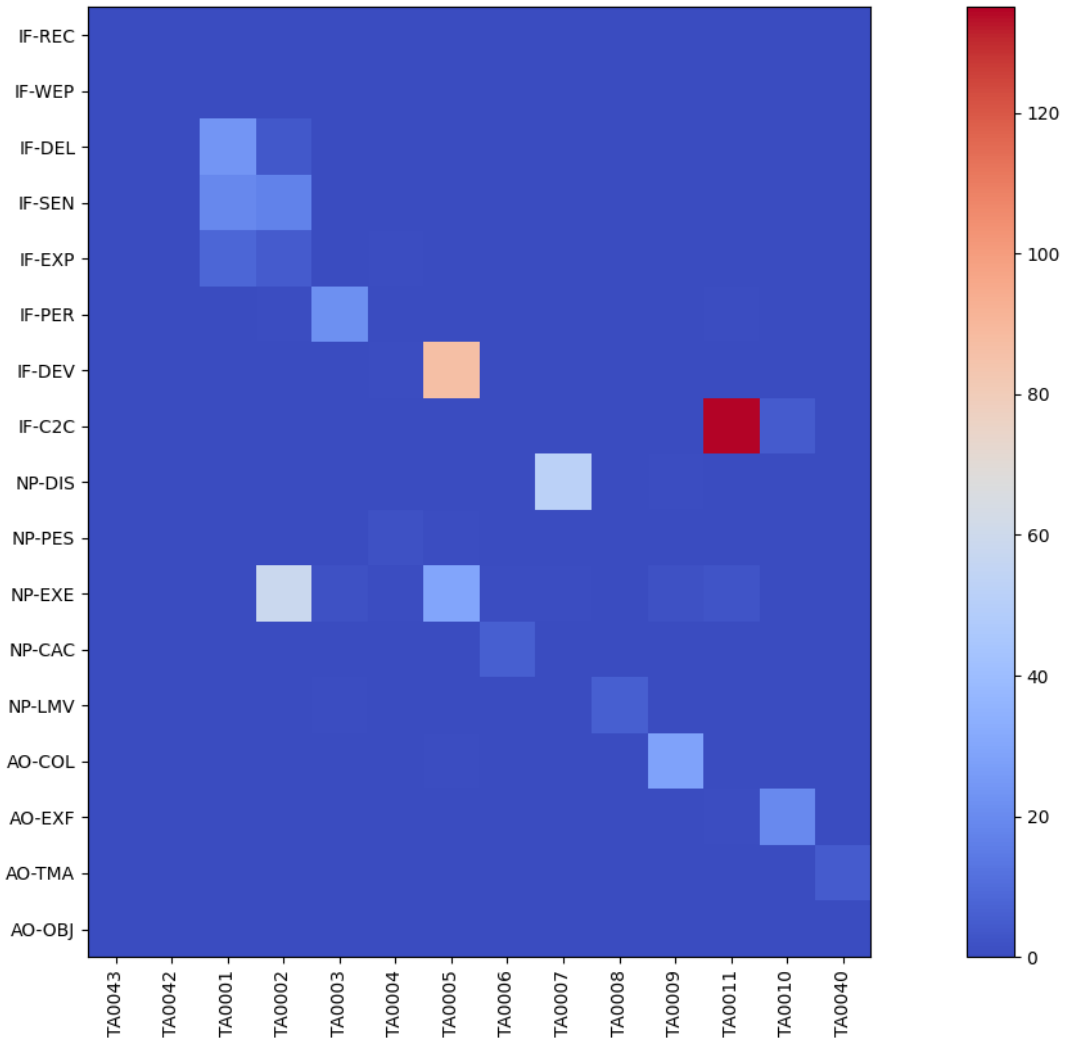


Figure 94 - UKC Step to ATT&CK Tactic Frequency Mapping Heatmap

The results below show a slightly refined view of above using percentages rather than counts. This provides a clearer presentation to show mapping between MITRE ATT&CK Tactic and UKC Phases (based on the reports analysed).

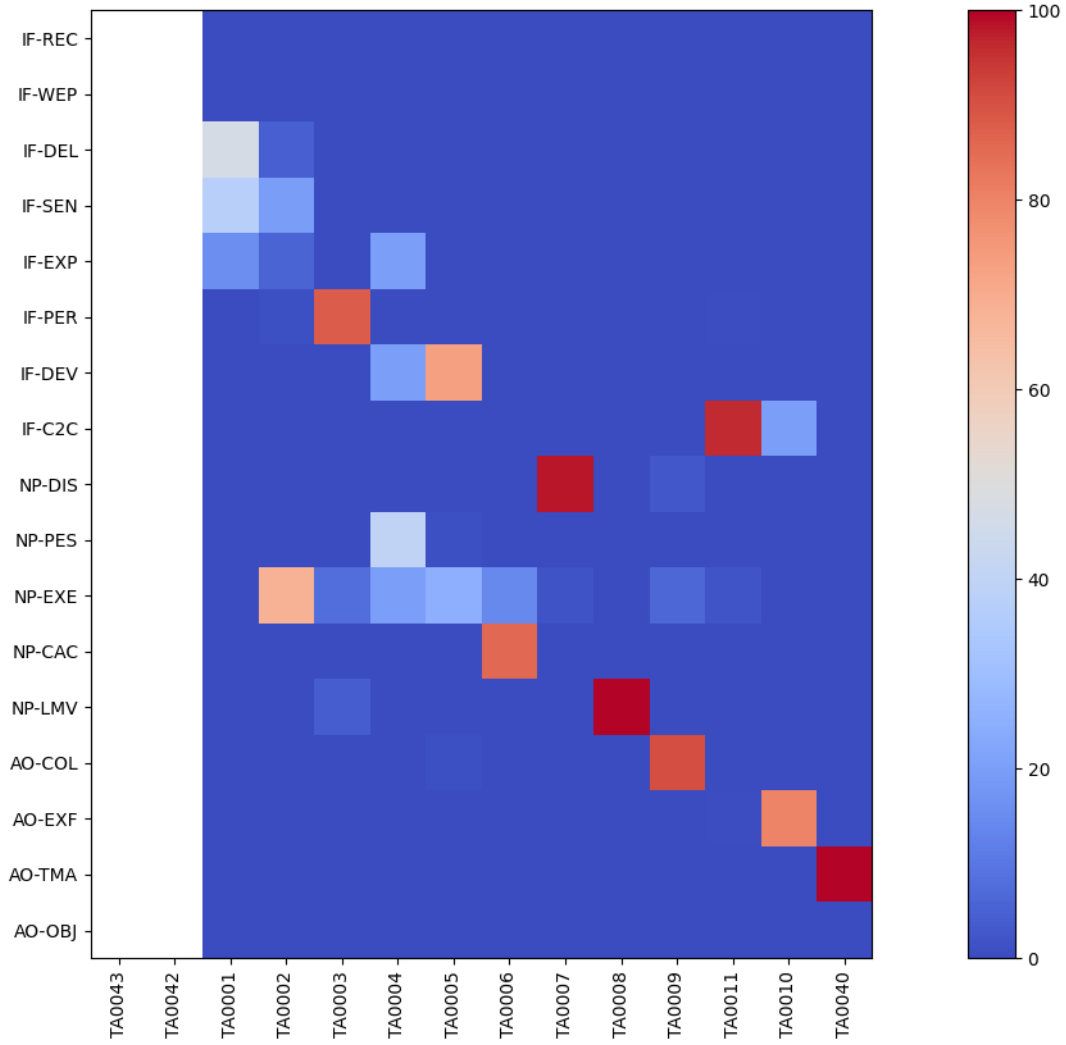


Figure 95 - Figure 86 - UKC Step to ATT&CK Tactic Frequency Mapping Col % Heatmap

The mapping between the ATT&CK Tactic and the UKC Steps is 'noisy' (i.e. often not a 1 to 1 mapping) and this is due to the granularity of both the UKC and ATT&CK element descriptions. This suggests some value in Future Work investigating how to make these more precise without reducing the usability of these models (see Future Work).

As mentioned previously TA0043 Reconnaissance and TA0042 Resource Development have not been used in these sequences as they are not (easily) detectable.

TA0001 – Initial Access

Maps onto UKC Delivery, Social Engineering and Exploitation (see also TA0002)

This makes sense as Exploitation elements are included under TA0001.

TA0002 - Execution

Maps primarily onto UKC Execution (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed. These may be seen as multiple UKC phases documented in the attack steps.

TA0003 - Persistence

Maps primarily onto UKC Persistence (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0004 – Privilege Escalation

Although this maps most frequently onto UKC Privilege Escalation as expected

However, in this case this is not as clearly dominant as above. This is probably due to the lower number of observations. Specific reference Privilege Escalation as an attack step is less common in the sample reports.

TA0005 – Defense Evasion

Maps primarily onto UKC Defence Evasion (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0006 – Credential Access

Maps primarily onto UKC Credential Access (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0007 – Discovery

Maps primarily onto UKC Discovery (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0008 – Lateral Movement

Maps only onto UKC Lateral Movement (as is to be expected)

TA0009 – Collection

Maps primarily onto UKC Collection (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0011 – Command & Control

Maps primarily onto UKC Command & Control (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0010 – Exfiltration

Maps primarily onto UKC Exfiltration (as is to be expected)

However, we also see mapping to other UKC phases. This is due to the granularity of both the UKC and ATT&CK technique descriptions where multiple elements can be assumed.

These may be seen as multiple UKC phases documented in the attack steps.

TA0040 – Impact

Maps only onto UKC Target Manipulation (as is to be expected)

This can now also be illustrated as follows

UKC Phase	UKC Step	Main associated ATT&CK Tactic
Initial Foothold		
	Reconnaissance	Out of scope
	Weaponization	Out of scope
	Delivery	TA0001 – Initial Access
	Social Engineering	TA0001 – Initial Access
	Exploitation	TA0001 – Initial Access
	Persistence	TA0003 - Persistence
	Defense Evasion	TA0005 – Defense Evasion
	Command & Control	TA0011 – Command & Control
Network Propagation		
	Discovery	TA0007 - Discovery
	Privilege Escalation	TA0004 – Privilege Escalation
	Execution	TA0002 - Execution
	Credential Access	TA0006 – Credential Access
	Lateral Movement	TA0008 – Lateral Movement
Action on Objectives		
	Collection	TA0009 - Collection
	Exfiltration	TA0010 - Exfiltration
	Target Manipulation	TA0040 - Impact
	Objectives	Not covered

Table 19 - Attacks as Tactics stream and UKC steps

Here also is an illustration of an attack cycle/sequence Lazarus_Group_002 as both a stream of Tactics and UKC steps. This also illustrates the comment above regarding granularity of both the MITRE Tactics and UKC steps. In this attack the attacker has sent an email with a malicious attachment. Following this the next step (TA0002) is for the recipient to click on the attachment. In this one step there are two elements. The recipient is tricked into thinking it is OK (undetectable) and then executes the attachment (detectable) for this reason this is mapped to two UKC steps in the one step.

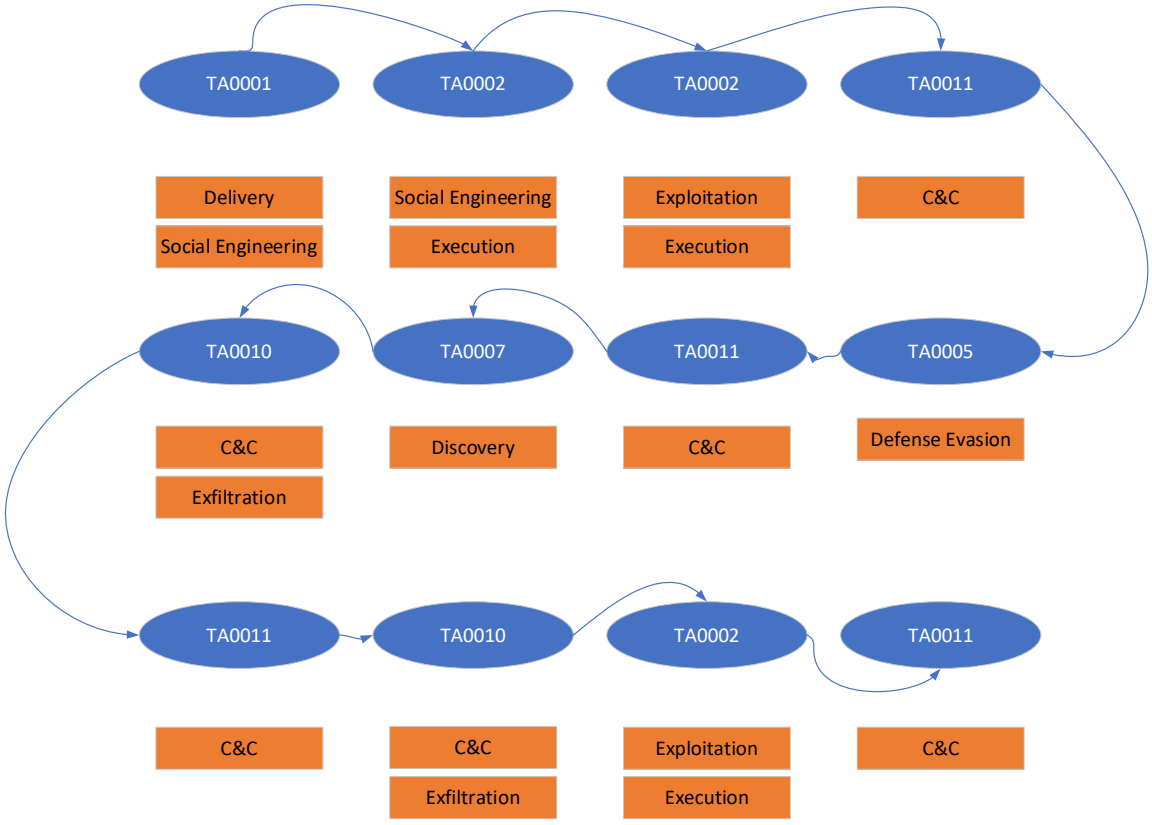


Figure 96 - Lazarus_Group_002 as Tactic and UKC stream/sequence

A deeper analysis of this observation about the mappings between ATT&CK and UKC alongside a wider sample of attack sequences is suggested in Future Work.

Chapter 8 Conclusion

8.1 Summary

This work has aimed at presenting an approach to improving the contribution that the ATT&CK knowledge base can provide to Cyber Situational Awareness. In terms of Endsley's high-level model this is in the areas of Comprehension and Projection. In terms of the CRUSOE model this is in Detection & Response.

It has approached this by considering the APT descriptions and recognising that additional intelligence can be provided by adding attack sequences. These sequences to be considered as an addition to the current given unordered list of Techniques, Tools, and Procedures that each of the APTs use.

This has been additionally motivated by the observation that in numerous research papers that there is a lack of availability of openly available intelligence data that can be used by researchers to test/investigate proposals. This work provides a proposal for creating a useful data set containing sequences of ATT&CK tactics and techniques derived from known reports (referenced in ATT&CK) of APT cyber-attacks.

To demonstrate an approach to tackling this, a number of open-source reports of attacks have been analysed and used to design a model that is used to create a 'ground truth' dataset of known attack sequence signatures. While collating this data we have found evidence in the reports to justify the inclusion of additional techniques for current MITRE ATT&CK APT descriptions.

To the authors knowledge this model had not previously been proposed in existing research.

These attack sequences have been converted into this model, which has been designed and tested using a representative set of APTs and several the attacks that have been attributed to them. The attacks recorded include examples across all the MITRE ATT&CK Tactics and the Unified Kill Chain phases. Approximately 100 different MITRE ATT&CK Techniques have been included in these attacks and multi-step attacks have been included to demonstrate chaining of these sequences.

To demonstrate the potential value of this data set, we have provided:

- A simple demonstration of how the sequences describing the attacks can provide more specific intelligence than the simple lists of techniques for the APTs

- A simple demonstration of how this data can be used to match observed technique sequences (in this case based on Longest Common Subsequence commonly used in genome sequencing etc)
- A simple demonstration of how this approach could develop further into the basis of a signature-based detection/mitigation planning system.
    - In this case using a Hidden Markov Model. The hidden states in this case are the probable Tactics associated with a stream of techniques.
    - A simple n-gram Markov Model is also demonstrated. Although the accuracy is limited this does demonstrate the differing behavioural patterns across the various APTs.
- A simple demonstration of how this approach can provide further input to future kill chain model research.

8.2 Limitations

See also future work proposals below

The most complete description of attack elements in the open-source data used seems to be around the Initial Foothold phase. This then has been the most strongly exercised area with less emphasis on subsequent phases.

It is unclear how to record repeated processes in the test data (e.g., keylogging tools). In real attacks, these will be seen as a part of much longer sequences with repeated execution technique patterns. Here the initial execution is recorded to provide a reasonable example.

The method of chaining sequences together is greatly simplified and some investigation into marking how sequences can feed back into each other may be beneficial.

The sequences have been derived from open-source reporting. Because the sequences have been built only from this information this does not present additional issues with the targets' privacy. However, to achieve more detailed examples with increased specificity about individual attack classification and detailed sequence steps may be constrained by such privacy issues.

8.3 Future Work

As documented above, a great deal of effort was expended creating, populating, and exercising the sequence model to provide the proposal here. Consideration of this model provided several proposals for future work and direction.

These are documented here, in no specific order. They also intended to provide further support for the potential value of development of this work.

Further Formalise the Attack Sequence Model

Consider formalisation of the model in terms of Finite State Machine.

Consider also an approach to grouping steps so multiple matches to observed sequences can be considered.

Pass Data To MITRE

The data analysed in this work could be passed to MITRE to provide additional input. Certainly, the sequences themselves are not currently included in the ATT&CK knowledge base.

Additional Group intelligence has been found in the reports collated within ATT&CK. This may well be due the evolution of the techniques defined within ATT&CK and a lack of resource available to revisit reports. This data could be usefully added.

It could also be that some of this intelligence is/was considered covered by tool techniques associated with the APTs, but this does not aid analysts understanding the specific behaviours of the APT groups and their attack structures.

Develop as Open-Source Knowledge Base

Development of this as an open-source knowledge base that integrates with and augments the current MITRE ATT&CK knowledge base. This should include further finishing development of the python ATT&CK loaders initially developed here.

This must include clear quality control measures, existing platforms such as Wikipedia and OpenStreetMap could provide invaluable use cases.

Develop Sequence Reporting Standards

Development of standards that could be embraced by cyber analysts while creating reports on attacks to encourage inclusion of attack sequencing in a machine-readable form (including review with expert users). The intention here would be for analysts to easily add these to the existing reports so that these elements of the reports can be more easily extracted automatically than is currently the case.

As mentioned in the sections above, NLP approaches are constrained in how such sequencing can be extracted. It is hoped that a simple standard would encourage analysts to document attacks in

a way that is efficient for themselves and allows easy further population of the above open knowledge base. The sequence intelligence is implicit in many of the reports and so easily documented. As the adoption of ATT&CK (hopefully) continues, early definition of such a standard would make its use more likely.

Specific Research into NLP Extraction of Sequences

As mentioned in the sections above, NLP approaches are constrained in how such sequencing can be extracted. However, complementary research into NLP approaches to improve gathering of attack sequences from openly available attack reports would be a useful area of investigation.

Add Timings to Improve Sequence Matching Precision

The model for recording attack sequences here does not currently include intelligence about timings (although this is included as an unpopulated item in the model). Further investigation and research are required. Generally, the open source CTI used here does not include detailed notes on timings (hence not being included at this point), however access to some example data would allow development and justification of this part of the model.

This would provide additional intelligence to be used in sequence matching (approaches to considering the timing element (e.g., observing and matching similar distributions of state lengths) within sequences have been developed and discussed within the social sciences in particular).

Use Data as Basis For Deeper Research Into Abstract Attack Patterns / Kill Chains

This Knowledge Base may be used as demonstrated in this work to provide additional insight into attacks detected and possible appropriate mitigation / courses of action. This Knowledge Base may be used to provide a ready resource (which currently seems to be lacking) into the structure and sequence of Attack Patterns (Tactics and Techniques used during malicious attacks)

Use Data to Motivate a More Formal Analysis of the ATT&CK Knowledge Base Structure

We would like to research more deeply the structure of the MITRE ATT&CK Tactics and Techniques. We have briefly discussed with the MITRE ATT&CK team, and it is clear they wish to keep the core knowledge base simple in structure. In particular they will only support two levels of technique hierarchy (although they also encourage local ‘extensions’ to the core ATT&CK model to tailor to individual needs). MITRE’s view is based on experience with CAPEC that has become very complicated and disjointed putting potential users off. However, we would like to

investigate how we could describe completeness of the scope provided by Tactic and Technique coverage and how relationships between elements could be better modelled. Examples include:

An attackers detailed behaviour at execution phase ensuring that all execution elements can be clearly expressed and sequenced. It is often possible to identify multiple techniques that are required to fully describe (based on information in the reports) a single functional step taken by an attacker.

> Reviewing description of lateral movement (e.g., if an attacker has accessed credentials and then using these to access remote system this can only be described as a combination of Initial Access and Lateral Movement techniques to illustrate). Perhaps clarifying where Lateral Movement sits in a multi-step attack model (is the end of one step or the start of another?).
>
> It seems beneficial to analyse the structure of the ATT&CK techniques more formally against attack models (including ontologies) to determine full coverage (and ideally unique mappings). This does not seem to currently exist and can be provided back to MITRE for consideration. It is understood that ATT&CKs success is based a pragmatic model that is easy to use by analysts. This then creating a 'common language'. Some similar observations have also recently been observed in (Naik et al., 2022) (see MITRE ATT&CK Advantages/Disadvantages)

<u>Development / Integration of More Sophisticated Classification Models</u>

Research into a more sophisticated classification model for this knowledge base. This thought is motivated by discussions such as in (Wang et al., 2020) and (Cai et al., 2020). Where APT intelligence modelling is investigated to enhance what can be deduced from the data available. This may include integration with more complex or specific classification models (such as (Villalón-Huerta et al., 2022) and (Ladisa et al., 2022)).

<u>Connecting Sequences to CAPEC, CWE and NVD</u>

Connecting this work with additional datasets such as CAPEC, CWE and NVD. There is certainly some interesting work currently with MITRE connecting vulnerabilities (CVE) with potential ATT&CK Techniques and Tactics (MITRE, 2022e). CAPEC attack patterns may also be added to the classification model where observed.

<u>Research Into Using This Data to Drive More Sophisticated Forward Prediction Models</u>

Investigation of more sophisticated machine learning approaches to forward attack step prediction. Recent examples include (T. Yu et al., 2022) which presents the seq2seq 'encoder-decoder' prediction model to estimate next steps in attack sequences based on previous attack sequence data. Hidden Markov Models also a common avenue investigation for further specific investigation (e.g. (Di Bernardino & Brogi, 2019) (Chadza et al., 2020) and (Dass et al., 2021) and more) although in (Ansari et al., 2020) evidence is presented to show deep learning models offer better ultimate performance.

Another specific area of further investigation includes Finite State Machines and whether these can be considered applicable in this case.

Research Into Using This Data to Drive More Sophisticated Forward Prediction Models

The examples provided here are very basic and included to illustrate potential. They are not intended as convincing models. However, this is an interesting area of research. NLP research includes examples of text prediction approaches (e.g., LSTM) but here the corpus is from multiple 'authors' with differing procedures. Additionally, we may possibly expect different sequence structures during different phases. For example, we may see consistency of command and control structure across an attack but discovery sequences may be tailored to individual circumstances.

Research Into How This Data Can Be Used to Support Red Teaming Automation

Data as input to red teaming automation. An approach is discussed in (T. Yu et al., 2022) but this presents some limitations on the attacker Tactic coverage. In (Elgh, 2022) adversary emulation tools are compared, the conclusions note that the manually created attack scenarios gave the least noisy results. This is to be expected as these document real world attacks. As this dataset builds it will provide an increasingly useful set of example attacks.

Research Into How This Data Can Be Used With Pedagogical Games

Data as input to future 'pedagogical games' For instance as described in 'Riskio: A Serious Game for Cyber Security Awareness and Education' (Hart et al., 2020). These sequences can be used to provide real attack examples to be 'played through'.

Extend Model to Allow Inclusion of Asset Intelligence Associated With Steps

Extending the model to include knowledge about 'Assets' related to attack steps. Defining a precise model for relevant asset identification is problematic (e.g., hardware, software, data, and appropriate level of detail), but could provide additional intelligence to identify most likely attack patterns. Although not investigated here the motivation for future inclusion is derived from the

reference to RiskIT in (Bodeau et al., 2018b) “A risk scenario is described in terms of threat type (which includes malicious threats), actor, type of event (i.e., type of impact), **asset or resource affected**, and time”.

Extend Interoperability of the Model

Formalise the attack model as a web ontology. The model as defined can be extended as a web ontology (RDF/OWL). This is an interesting area to explore as it may lead to increased analytical and sharing opportunities.

As above implementing this model in STIX-like objects to further aid integration. Future versions of this would benefit better from being converted into JSON encoding of STIX (including customisation). As well as conforming to open standards this would support stronger integration with the ATT&CK dataset.

Additional fields for Steps:

> Exploits – CVE/CWE reference to aid integration with vulnerability/weakness databases (this could also include knowledge of CAPEC patterns).
>
> Asset – Knowledge about the asset being attacked (see also above)
>
> Confidence – Confidence in the ‘knowledge’ to support more nuanced decision-making models

Deeper Analysis of Mappings Between ATT&CK Tactic Model and UKC

Further analysis of the UKC mapping. Initial basic statistics showing alignment of Tactics and UKC elements would be useful. Also, in creating this model, I have found some areas of vagaries when interpreting how an attack maps onto the Unified Kill Chain (UKC). This could be investigated further to improve the fit between the UKC described and MITRE ATT&CK. This is mainly around how the relationship between the Initial Foothold and Network Propagation phases describe the attack. Examples include:

> How execution is described. The attacker will often execute code to complete installation and persistence yet ‘Execution’ is described in the Network Propagation phase.
>
> Some MITRE ATT&CK Defense Evasion techniques (such as T1218.011 : System Binary Proxy Execution: Rundll32) also imply execution. This leads to some confusion over Initial Foothold and Network Propagation phases.

Following this a further study of the data could be conducted to understand relationships between ATT&CK Tactic use and various Kill Chain stages.

Improve Flexibility of Chaining Models

Currently the model supports the ability to chain attack sequences together to model lateral movement. Consider adding an additional field to show if a Technique being actioned within a sequence is caused by another attack sequence. Where a pivot is created it is possible for the attack sequence following lateral movement to cause events in an attack sequence associated with the pivot point. Currently these are seen as another sequential event

The event id numbering is currently clumsy to implement. Would be better to remove the id line and generate on load. The Prev id could be calculated automatically through analysis of G/S value.

Develop Supporting Toolkits

Toolkit to help ease creation of sequences. Some standard technique/kill chain sequences were observed and could be offered as 'click in' options to aid speed of construction.

Refine Attack Sequence Matching Model Further

Further consideration of sequence matching

The Command & Control sequences may themselves provide a strong indication of matching behaviour.

Detailed differences in sections of the order of Discovery techniques may obscure strength of matches.

In the discussion above an attempt is made to provide some protection from detailed differences in matches by including consideration of matching at main technique level as well as the more detailed sub technique level. However, specific details of techniques such as those used during Initial Access may be more important in matching attacks. For instance the knowledge of the precise Phishing (T1566) approach may be more valuable than Account Discovery (T1087) decisions.

Integrating Meta-Data model with more complex and specific models

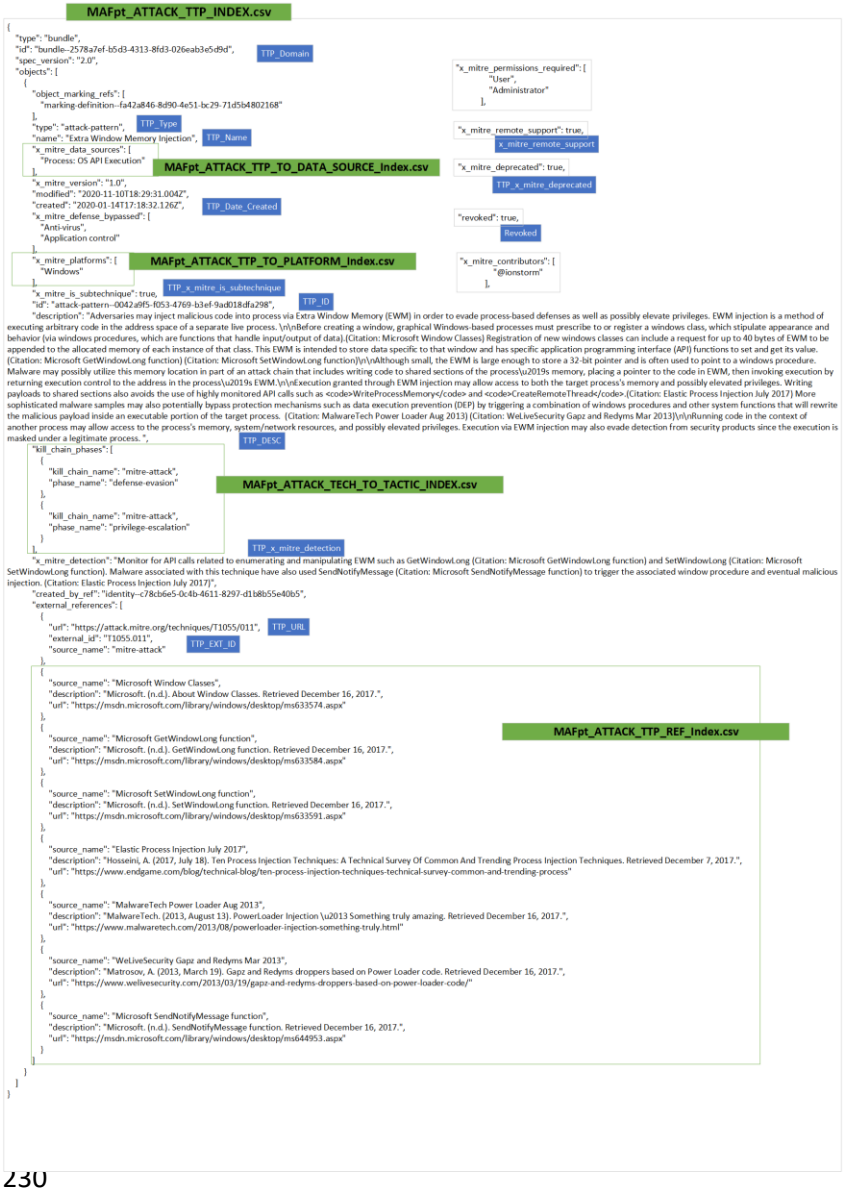
Consideration could be given to integrating the Meta-Data model with more complex and specific models. For instance in (Ladisa et al., 2022) the authors seek to define a taxonomy for attacks on open-source software supply chains. Using an Attack Graph type approach, the authors describe 107 different attack vector types in this domain. These specific vectors could be used to

complement the MITRE ATT&CK Initial Access Technique of T1195 – Supply Chain Compromise (or one of the three more detailed sub techniques).

Appendix A MAFpt – ATT&CK Relational Model (based on V10.1)

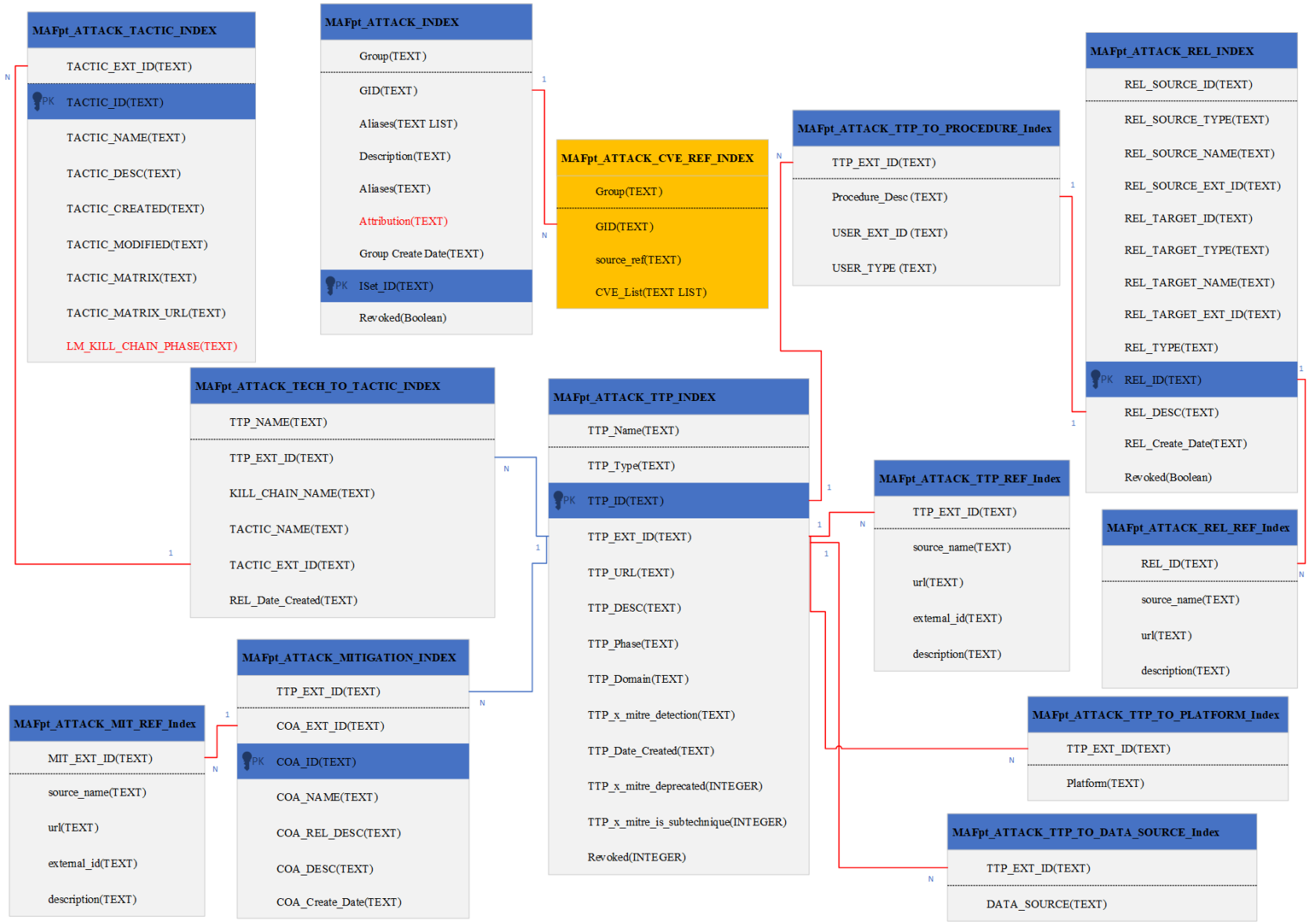
A.1 Overview

An example attack-pattern in STIX2/JSON form


MAFpt_ATTACK_TTP_INDEX.csv
TTP_Domain
TTP_Type
TTP_Name
MAFpt_ATTACK_TTP_TO_DATA_SOURCE_Index.csv
TTP_Date_Created
MAFpt_ATTACK_TTP_TO_PLATFORM_Index.csv
TTP_x_mitre_is_subtechnique
TTP_ID
TTP_DESC
MAFpt_ATTACK_TECH_TO_TACTIC_INDEX.csv
TTP_x_mitre_detection
TTP_URL
TTP_EXT_ID
MAFpt_ATTACK_TTP_REF_Index.csv
x_mitre_remote_support
TTP_x_mitre_deprecated
Revoked

```
{
   "type": "bundle",
   "id": "bundle--46cda918-31e4-4bd4-bbea-82fc8e00ea4a",
   "spec_version": "2.0",
   "objects": [
      {
         "object_marking_refs": [
            "marking-definition--fa42a846-8d90-4e51-bc29-71d5b4802168"
         ],
         "modified": "2021-11-10T09:30:48.753106Z",
         "id": "relationship--9567076b-2a77-43e4-befd-19556def9d47",
         "target_ref": "attack-pattern--910906dd-8c0a-475a-9cc1-5e029e2fad58",
         "source_ref": "x-mitre-data-component--3d20385b-24ef-40e1-9f56-f39750379077",
         "created_by_ref": "identity--c78cb6e5-0c4b-4611-8297-d1b8b55e40b5",
```

```
        "relationship_type": "detects",
        "created": "2021-11-10T09:30:48.753106Z",
        "type": "relationship"
      }
  ]
}
```


MAFpt_ATTACK_TACTIC_INDEX
TACTIC_EXT_ID(TEXT)
PK TACTIC_ID(TEXT)
TACTIC_NAME(TEXT)
TACTIC_DESC(TEXT)
TACTIC_CREATED(TEXT)
TACTIC_MODIFIED(TEXT)
TACTIC_MATRIX(TEXT)
TACTIC_MATRIX_URL(TEXT)
LM_KILL_CHAIN_PHASE(TEXT)
MAFpt_ATTACK_INDEX
Group(TEXT)
GID(TEXT)
Aliases(TEXT LIST)
Description(TEXT)
Aliases(TEXT)
Attribution(TEXT)
Group Create Date(TEXT)
PK ISet_ID(TEXT)
Revoked(Boolean)
MAFpt_ATTACK_CVE_REF_INDEX
Group(TEXT)
GID(TEXT)
source_ref(TEXT)
CVE_List(TEXT LIST)
MAFpt_ATTACK_TTP_TO_PROCEDURE_Index
TTP_EXT_ID(TEXT)
Procedure_Desc (TEXT)
USER_EXT_ID (TEXT)
USER_TYPE (TEXT)
MAFpt_ATTACK_REL_INDEX
REL_SOURCE_ID(TEXT)
REL_SOURCE_TYPE(TEXT)
REL_SOURCE_NAME(TEXT)
REL_SOURCE_EXT_ID(TEXT)
REL_TARGET_ID(TEXT)
REL_TARGET_TYPE(TEXT)
REL_TARGET_NAME(TEXT)
REL_TARGET_EXT_ID(TEXT)
REL_TYPE(TEXT)
PK REL_ID(TEXT)
REL_DESC(TEXT)
REL_Create_Date(TEXT)
Revoked(Boolean)
MAFpt_ATTACK_TECH_TO_TACTIC_INDEX
TTP_NAME(TEXT)
TTP_EXT_ID(TEXT)
KILL_CHAIN_NAME(TEXT)
TACTIC_NAME(TEXT)
TACTIC_EXT_ID(TEXT)
REL_Date_Created(TEXT)
MAFpt_ATTACK_TTP_INDEX
TTP_Name(TEXT)
TTP_Type(TEXT)
PK TTP_ID(TEXT)
TTP_EXT_ID(TEXT)
TTP_URL(TEXT)
TTP_DESC(TEXT)
TTP_Phase(TEXT)
TTP_Domain(TEXT)
TTP_x_mitre_detection(TEXT)
TTP_Date_Created(TEXT)
TTP_x_mitre_deprecated(INTEGER)
TTP_x_mitre_is_subtechnique(INTEGER)
Revoked(INTEGER)
MAFpt_ATTACK_TTP_REF_Index
TTP_EXT_ID(TEXT)
source_name(TEXT)
url(TEXT)
external_id(TEXT)
description(TEXT)
MAFpt_ATTACK_REL_REF_Index
REL_ID(TEXT)
source_name(TEXT)
url(TEXT)
description(TEXT)
MAFpt_ATTACK_MITIGATION_INDEX
TTP_EXT_ID(TEXT)
COA_EXT_ID(TEXT)
PK COA_ID(TEXT)
COA_NAME(TEXT)
COA_REL_DESC(TEXT)
COA_DESC(TEXT)
COA_Create_Date(TEXT)
MAFpt_ATTACK_MIT_REF_Index
MIT_EXT_ID(TEXT)
source_name(TEXT)
url(TEXT)
external_id(TEXT)
description(TEXT)
MAFpt_ATTACK_TTP_TO_PLATFORM_Index
TTP_EXT_ID(TEXT)
Platform(TEXT)
MAFpt_ATTACK_TTP_TO_DATA_SOURCE_Index
TTP_EXT_ID(TEXT)
DATA_SOURCE(TEXT)

Appendix B MAFpt – Brief Summary Of Python Code

B.1 Overview

B.2 Utility Functions

```
#
#   def GetListOfGroups(self) - return(Group_list)
#   def GetGroupName(self,
#                        GID) - return(str(GroupNameStr))
#   def GetGroupAttribution(self,
#                        GroutAttribution) - return(str(GroutAttributionStr))
#   def GetListOfTactics(self) - return(Tactic_list)
#    def GetListOfTacticsInDomain(self,
#                  [enterprise-attack, mobile-attack, pre-attack] DomainName) - return(Tactic_list)
#   def GetListOfKCPhasesForTactic(self,
#                                       tactic_ext_id): - return(KC_Phase_list)
#   def GetListOfTacticsWithMatrix(self) - return(ListOfDicts)
#                                           dict_row={TACTIC_EXT_ID"],
#                                                   "TACTIC_MATRIX"]}
#   def GetListOfTTP(self) - return(TTP_list)
#   def GetListOfTTPInDomain(              self,
#      [Enterprise|Mobile|Pre-ATT&CK ]    domain) - return(TTP_list)
#   def GetListOfTTPOfType(              self,
#      [attack-pattern|malware|tool]     type) - return(TTP_list)
#   def GetTTPForGROUP(self,
#                           group_name) - return(TTPCurrentList)
#   def GetTTPForGROUPByLevel(self,
#                           group_name,
#      [ 'Top'|'Sub']     Level,
#      ["Y"|"N"]              GetTopFromSub) - return(TTPCurrentList)
#   def GetTTPDomain(self,
#                        ThisTTP) - return(Type)
#   def GetTTPType(self,
#                        ThisTTP) - return(Domain)  [attack-pattern|malware|tool]
#   def GetTTPLevel(self,
#                        ThisTTP) - return(Level) [ 'Top' or 'Sub']
#   def GetTacticsForGROUP(self,
#                           group_name) - return(GroupTacticList)
#
```

```
#    def GetTacticsForTTP(self,
#                         ThisTTP) - return(TacticList)
#    def GetTacticName(self,
#                          Tactic_Ext_Id) - return(TacticName)
#    def GetTacticDomain(self,
#                           Tactic_Ext_Id) - return(TacticDomain)
#                                 return(enterprise-attack|mobile-attack|pre-attack)
#    def GetSchema(self) - prints out dataframe structures
#
#    def GetTOOL_TTPDetails(self,
#                                  TOOL_id)
#       return - TTP_Details=self.ATTACK_TTP_Index[(self.ATTACK_TTP_Index['TTP_EXT_ID'] ==
TTP_id)].to_dict(orient='records')[0])
#    def GetTTPDetails(self,
#                           TTP_id)
#       return - TTP_Details=self.ATTACK_TTP_Index[(self.ATTACK_TTP_Index['TTP_EXT_ID'] ==
TTP_id)].to_dict(orient='records')[0]
#    def GetGROUPDetails(self,
#                           GROUP_id):
#       return - GROUP_Details=self.ATTACK_Index[(self.ATTACK_Index['Group'] ==
GROUP_id)].to_dict(orient='records')[0]
#     def GetGroupTTPRelDate(self,
#                                GROUP_NAME,
#                                TTP) - return(rel_date)
#     def GetToolTTPRelDate(self,
#                                Tool_TTP,
#                               TTP) - return(rel_date)
#     def SelectTOOL_TTP(self,
#                           TTPList) - return(TOOL_TTPList)
#      def GetTTPForTOOL(self,
#                          tool_name):- return(TTP_List)
#      def GetReportsForTech(self,
#                                TTP): - return(TTP_List)
#      def GetMitigationsForTech(self,
#                                TTP): - return(TTP_List)
#       def GetTechForRef(self,
#                                REF): - return(TTPList)
#       def GetTacticForRef(self,
#                                REF): - return(TacticList)
#       def GetTTPRefCounts(self,
#                                ): - return(Series)
#        def GetListOfTTPReferences(self
```

```
#                                ):- return(TTP_URL_list)
#       def GetTacticForRef(self,
#                                REF):  -  return(Tactic_list)
#       def GetProcTextForTTP(self,
#                                TTP_EXT_ID): - return(ProcTextList)
#       def GetCountsTacticUseByGroup(self,
#                  [enterprise-attack, mobile-attack, pre-attack] Matrix): - return(RefCount)
#

#
```

Appendix C Manual Example Attack Sequences

C.1 Overview

These are provided as detailed background to attack sequence model analysis (a larger set has been produced for the full test set described above). Once the initial tables were produced these were converted to CSV files. Then loaded into a graph model using python.

Each example includes references to the related attack reports and supporting intelligence.

C.2 admin@338

Dimension	***Technique***	***Notes***
Attribution	admin@338	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	China	
Target Location	Hong Kong	
Target Type	Media	
Impact	Exfiltration	Monitoring media orgs
Vulnerabilities Exploited	CVE-2012-0158	
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new Sequence models
Date	2015	

	Sequence_ID	***admin@338_001***		***Ver***	***0.1***		

ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	[1] Spearphishing - Hong Kong based organisations
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	**[1] Includes malicious file (trojan downloader Lowball [2])**
3	**TA0002 : Execution**	**T1203 : Exploitation for Client Execution**	2		S	IF-EXP	**[1] The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code)**
4	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	3		G	IF-C2C	This technique to provides more detail on step below
5	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	3		S	IF-C2C	[1] Initial installation connects to C&C
6	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	5		S	IF-C2C/IF-PER	[1] Install upgraded tool
7	**TA0002 : Execution**	**T1059.003 Command and Scripting Interpreter: Windows Command Shell**	6		S	NP-EXE	[1] Still part of Initial Access step. Commands executed via .bat. Discovery

							Now UKC Pivot Occurs
8	**TA0007 Discovery**	**T1083 File and Directory Discovery**	7		S	NP-DIS	[1]
9	**TA0007 Discovery**	**T1082 System Information Discovery**	8		S	NP-DIS	[1]
10	**TA0007 Discovery**	**T1016 System Network Configuration Discovery**	9		S	NP-DIS	[1]
11	**TA0007 Discovery**	**T1007 System Service Discovery**	10		S	NP-DIS	[1]
12	**TA0007 Discovery**	**T1069.001 Permission Groups Discovery: Local Groups**	11		S	NP-DIS	[1]
13	**TA0007 Discovery**	**T1049 System Network Connections Discovery**	12		S	NP-DIS	[1]
14	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	13		G	IF-C2C	[1] This technique to provide more detail on step below
15	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	13		S	IF-C2C	[1] [6] Initial installation connects to C&C
16	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	15		S	IF-C2C	[1] Install second stage tool (Bubblewrap)

17	**TA0002 : Execution**	**T1059.003 Command and Scripting Interpreter: Windows Command Shell**	16		S	NP-EXE/IF-PER	To install second stage tool above
		Techniques unclear here					We know [2] The BUBBLEWRAP malware is installed with admin rights and the threat actors gain full access to the compromised machine [2] The BUBBLEWRAP Trojan may create a hidden system administrator account.
	TA005 Defense Evasion	**T1036.005 Masquerading: Match Legitimate Name or Location**	15		G		Rename second stage tool with benign name. But this is on the DropBox server
18	**TA007 Discovery**	**T1082 System Information Discovery**	17		S	NP-DIS	[1]
19	**TA0011 : Command & Control**	**T1102.002 Web Service: Bidirectional Communication**	18		S	IF-C2C	[6] Bubblewrap communications (not via Dropbox server, this Tech noted in ATT&CK)

[1] China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets | Mandiant [2015]

[2] BUBBLEWRAP Trojan Removal Report (enigmasoftware.com) (via Google - a little more clarity on Lowball and Bubblewrap load sequence). [????]

[3] Y-Security performs Attack Simulations, Penetration Tests, and Security Trainings (via Google a restatement of ATT&CK data) [????]

[4] Malware That Hides C&C Server on Dropbox Detected in the Wild (softpedia.com) (via Google - a little more clarity on Bubblewrap) [2015]

[5] (PDF) State-of-the-Art in Chinese APT Attack and Using Threat Intelligence for Detection. A Survey (researchgate.net) (a high level summary, again a restatement of ATT&CK data) [2022]

[6] BUBBLEWRAP, Software S0043 | MITRE ATT&CK®

C.3 APT28_001

Dimension	***Technique***	***Notes***
Attribution	APT28	
Initial Access Vector	T1078 : Valid Accounts	
Attack Origin	Russia	
Target Location	United States	[1]
Target Type	Government	[1]
Impact	Exfiltration	
Vulnerabilities Exploited	CVE-2020-0688, CVE 2020-17144	[1]
Related Attack Patterns		TBC
Preceded By	TBC	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2021	

	Sequence_ID	*APT28_001*		*Ver*	*0.1*		
ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*
1	**TA0006 : Credential Access**	**T1110 : Brute Force**	0		G	NP-CAC	[1] brute force capability allows the actors to access protected data, including email, and identify valid account credentials Info w.r.t. this attack scope
2	**TA0001 : Initial Access**	**T1133 : External Remote Services**	0		G	IF-EXP	[6] APT28 has used Tor and a variety of commercial VPN services to route brute force authentication attempts Info w.r.t to this attack scope
3	**TA0001 : Initial Access**	**T1078 : Valid Accounts**	0		S	IF-EXP/IF-DEL	[1] Use valid creds for initial access combine with CVEs…. Potential here to also reference weaponization tactic but as this

							the actual initial access then described as exploitation of the passwords. The ATT&CK Resource Development Tactic is unclear on this.
4	**TA0001 : Initial Access**	**T1190 : Exploit Public-Facing Application**	3		S	IF-EXP	[1] The actors used a variety of public exploits, including CVE 2020-0688 and CVE 2020-17144 to gain privileged remote code execution on vulnerable Microsoft Exchange servers. In some cases, this exploitation occurred after valid credentials were identified by password spray, as these vulnerabilities require authentication as a valid user

5	**TA0005 : Defense Evasion**	**T1027 : Obfuscated Files or Information**	4		G	IF-DEV	[1] The actors used a modified and obfuscated version of the reGeorg web shell to maintain persistent access on a target's Outlook Web Access (OWA®) server.
6	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	4		S	IF-C2C	[9] The NSA says that once they gain access, they will spread laterally through the network while deploying a reGeorg web shell for persistence, harvesting other credentials, and stealing files
7	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	6		S	IF-C2C	To web shell
8	**TA0002 : Execution**	**T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**	7		S	NP-EXE	[1] The actors used the ntdsutil.exe utility, which was present on a target's Active Directory® server

							to export the Active Directory database for credential access
9	**TA0006 : Credential Access**	**T1003.003 : OS Credential Dumping: NTDS**	8		S	NP-CAC	[1] The actors used the ntdsutil.exe utility, which was present on a target's Active Directory® server to export the Active Directory database for credential access
10	**TA0010 : Exfiltration**	**T1567 : Exfiltration Over Web Service**	9		S	AO-EXF	See above

[1] T1133/T1078 CSA_GRU_GLOBAL_BRUTE_FORCE_CAMPAIGN_UOO158036-21.PDF (defense.gov) Russian GRU Conducting Global Brute Force Campaign to Compromise Enterprise and Cloud Environments [2021]

[2] STRONTIUM: Detecting new patterns in credential harvesting - Microsoft Security Blog From [1] [2020]

[3] T1078 – Two Years of Pawn Storm Examining an Increasingly Relevant Threat (trendmicro.com) [2017]

[4] T1078 - Indictment (justice.gov) [2018]

[5] T1078 - Corporate IoT – a path to intrusion – Microsoft Security Response Center [2019]

[6] APT28, IRON TWILIGHT, SNAKEMACKEREL, Swallowtail, Group 74, Sednit, Sofacy, Pawn Storm, Fancy Bear, STRONTIUM, Tsar Team, Threat Group-4127, TG-4127, Group G0007 | MITRE ATT&CK®

[7] GitHub - mandiant/iocs: FireEye Publicly Shared Indicators of Compromise (IOCs)

[8] reGeorg (Malware Family) (fraunhofer.de) (via search on reGeorg)

[9] NSA: Russian GRU hackers use Kubernetes to run brute force attacks (bleepingcomputer.com) [2021]

[10] Black Hat Europe 2014 | Arsenal regeorg info [2014]

C.4 APT28_002

Dimension	*Technique*	*Notes*
Attribution	APT28	
Initial Access Vector	T1566.002 : Spearphishing Link	
Attack Origin	Russia	
Target Location	United States	[1]
Target Type	Government	[1]
Impact	Exfiltration	
Vulnerabilities Exploited		[1]
Related Attack Patterns		TBC
Preceded By	TBC	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2016	

	Sequence_ID	*APT28_002*		*Ver*	*0.1*		
ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*
1	**TA0001 : Initial Access**	**T1566.002 : Phishing: Spearphishing Link**	0		S	IF-DEL/IF-SEN	[1] p6 21 created and sent a spearphishing email to the

							chairman of the Clinton Campaign ….
							Credential access achieved on external system (via link)
2	**TA0001 : Initial Access**	**T1078 : Valid Accounts**	1		S	IF-EXP	[1] … instructing the user to change his password by clicking the embedded link. Those instructions were followed
3	**TA0009 : Collection**	**T1114 : Email Collection**	2		S	AO-COL	[1] On or about March 21, 2016, the co-conspirators stole the contents of the chairman's email account, which consisted of over 50,000 emails

[1] Indictment (justice.gov) [2018]

C.5 APT28_003

Dimension	***Technique***	***Notes***
Attribution	APT28	

Initial Access Vector	T1566.002 : Spearphishing Link	
Attack Origin	Russia	
Target Location	United States	[1]
Target Type	Government	[1]
Impact	Exfiltration	
Vulnerabilities Exploited		[1]
Related Attack Patterns		TBC
Preceded By	TBC	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2016	

	Sequence_ID	***APT28_003***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.002 : Phishing: Spearphishing Link**	0		S	IF-DEL/IF-SEN	[1] 24 Employee 1 had received a spearphishing email from the Conspirators on or about April 6, 2016
							Credential access achieved on external system (via link)

2	**TA0001 : Initial Access**	**T1078 : Valid Accounts**	1		S	IF-EXP	[1] … , and entered password after clicking on the link.
3	**TA0011 : Command and Control**	**T1105 : Ingress Tool Transfer**	2		S	IF-C2C	[1] Between in or around April 2016 and June 2016, the Conspirators installed multiple versions of their X-Agent malware on at least ten DCCC computers, which allowed them to monitor individual employees' computer activity, steal passwords, and maintain access to the DCCC network
4	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	3		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In
5	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	4		S	AO-COL	[1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees

6	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	5		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In
7	**TA0009 : Collection**	**T1113 : Screen Capture**	6		S	AO-COL	[1] April 14, 2016, the Conspirators repeatedly activated X-Agent's keylog and screenshot functions to surveil DCCC Employee 1's computer activity over the course of eight hours
8	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	7		S	AO-COL	[1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees
9	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	8		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In
10	**TA0009 : Collection**	**T1113 : Screen Capture**	9		S	AO-COL	[1] April 14, 2016, the Conspirators repeatedly activated X-Agent's keylog and screenshot

							functions to surveil DCCC Employee 1's computer activity over the course of eight hours
11	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	10		S	AO-COL	[1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees Includes credential access at this point
12	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	11		S	AO-COL	[1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees Includes credential access at this point
13	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	12		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In

[1] Indictment (justice.gov) [2018]

[2] XAgentOSX, Software S0161 | MITRE ATT&CK® (ATT&CK)

[3] XAgentOSX: Sofacy's XAgent macOS Tool (paloaltonetworks.com) (via Google [2017])

C.6 APT28_004

Dimension	*Technique*	*Notes*
Attribution	APT28	
Initial Access Vector	T1078 : Valid Accounts	
Attack Origin	Russia	
Target Location	United States	[1]
Target Type	Government	[1]
Impact	Exfiltration	
Vulnerabilities Exploited		[1]
Related Attack Patterns		TBC
Preceded By	APT28_003	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2021	

	Sequence_ID	*APT28_004*		*Ver*	*0.1*		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1078 : Valid Accounts**	0		S	IF-EXP	[1] 26 Use valid creds gained in APT28_003 for initial access

2	**TA0008 : Lateral Movement**	**T1570 : Lateral Tool Transfer**	1		S	NP-LMV	[1] 26 In or around April 2016, the Conspirators installed X-Agent malware on the DNC network, including the same versions installed on the DCCC network.
3	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	2		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In
4	**TA0002 : Execution**	**T1059.001 : <Command and Scripting Interpreter>:PowerShell**	3		S	NP-EXE	[3] XAgent has ability to receive commands from threat actors via its command and control channel [2] APT28 downloads and executes PowerShell scripts and performs PowerShell commands.
5	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	4		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In

6	**TA0009 : Collection**	**T1113 : Screen Capture**	5		S	AO-COL	[1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers
7	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	6		S	AO-COL	[1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers
8	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	7		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In
9	**TA0009 : Collection**	**T1113 : Screen Capture**	8		S	AO-COL	[1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers
10	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	9		G	AO-COL	[1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers [1] such as a screenshot and keystroke capture of Employee 2 viewing the online

							banking information
11	**TA0011 : Command and Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	9		S	IF-C2C	[2] [3] XAgent uses HTTP requests to communicate with its C2 servers Message Out/In
12	**TA0011 : Command and Control**	**T1105 : Ingress Tool Transfer**	11		S	IF-C2C	[1] 28 To enable them to steal a large number of documents at once without detection, the Conspirators used a publicly available tool to gather and compress multiple documents on the DCCC and DNC networks.
13	**TA0002 : Execution**	**T1059.001 : <Command and Scripting Interpreter>:PowerShell**	12		S	NP-EXE	[2] APT28 has retrieved internal documents from machines inside victim environments, including by using Forfiles [3] to stage documents before exfiltration
14	**TA0009 : Collection**	**T1005 : Data from Local System**	13		S	AO-COL	[2] APT28 has retrieved internal documents from machines inside

							victim environments, including by using Forfiles [3] to stage documents before exfiltration
15	**TA0011 : Command and Control**	**T1573.001 : <Encrypted Channel>:Symmetric Cryptography**	14		S	IF-C2C	[1] The Conspirators then used other GRU malware, known as "X-Tunnel," to move the stolen documents outside the DCCC and DNC networks through encrypted channels. [2] XTunnel uses SSL/TLS and RC4 to encrypt traffic.

[1] Indictment (justice.gov) [2018]

[2] XAgentOSX, Software S0161 | MITRE ATT&CK®

[3] XAgentOSX: Sofacy's XAgent macOS Tool (paloaltonetworks.com) (via Google [2017])

[3] Forfiles, Software S0193 | MITRE ATT&CK®

C.7 APT29_001

Dimension	***Technique***	***Notes***
Attribution	APT29	

Initial Access Vector	T1566.001 : Spearphishing Attachment	[6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro
Attack Origin	Russia	Via ATT&CK/TCERT
Target Location	US	[6] The Dukes launched several waves of highly targeted spear phishing attacks against several U.S.-based think tanks and NGOs
Target Type	U.S.-based think tank	[6] See above
Impact	Exfiltration (confidentiality)	
Vulnerabilities Exploited	CVE-2021-36934	
Related Attack Patterns		
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2016	[6] Wave 2

	Sequence_ID	***APT29_001***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL	[2] In the next evolution of the campaign, MSTIC observed NOBELIUM

							attempting to compromise systems through an HTML file attached to a spear-phishing email
2	TA0005 : Defense Evasion	T1027.006 : <Obfuscated Files or Information>:HTML Smuggling	1		G	IF-DEV	[2] See above/below
3	TA0002 : Execution	T1204.002 : User Execution : Malicious File	1		S	IF-SEN/NP-EXE	[2] User executes HTML malicious HTML file
4	TA0002 : Execution	T1059.007 : Command and Scripting Interpreter: JavaScript	3		S	NP-EXE/IF-SEN	[2] When opened by the targeted user, a JavaScript within the HTML wrote an ISO file to disc and encouraged the target to open it
5	TA0005 : Defense Evasion	T1553.005 : Subvert Trust Controls: Mark-of-the-Web Bypass	4		G	IF-DEV	[2] As above ISO embedded in HTML
6	TA0005 : Defense Evasion	T1480 Execution Guardrails	4		G	IF-DEV	[2] The actor sometimes employed checks for specific internal Active Directory domains that would terminate execution of the malicious process if it identified an

							unintended environment.
7	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	4		S	IF-C2C	[2] Cobalt Strike Beacon begins communication
8	**TA0004 : Privilege Escalation**	**T1068 : Exploitation for Privilege Escalation**	7		S	NP-PES	[3] CVE-2021-36934 See also [7]
9	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	8		S	IF-C2C	[2] Communication leading to installation of Adfinder
10	**TA0011 : Command & Control**	**T1105 : Ingress Tool Transfer**	9		S	IF-C2C	[3] Install Adfinder - A tool to query the Active Directory
11	**TA0007 : Discovery**	**T1087.002 : <Account Discovery>:Domain Account**	10		S	NP-DIS	[3] Once attackers have a foothold on the machine, they usually deploy additional tools to gather information about the host system or other machines in the same network
12	**TA0007 : Discovery**	**T1082 : System Information Discovery**	11		S	NP-DIS	[3] See above
13	**TA0009 : Collection**	**T1213 : Data from Information Repositories**	12		S	AO-COL	[3] Attacker able to collect data via this account

14	**TA0010 : Exfiltration**	**T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol**	13		S	AO-EXF	Exfiltration of data (following above). Known procedure from [8]
15	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	14		S	IF-C2C	[2] Communication leading to installation of Sharp-SMBExec
16	**TA0011 : Command & Control**	**T1105 : Ingress Tool Transfer**	15		S	IF-C2C	[3] Install Sharp-SMBExec - A tool to execute a command on a remote machine using SMB (this will be used to achieve lateral movement in APT29_004). Similar behaviour / tool usage also noted in use of PSExec [8]
17	**TA0010 : Exfiltration**	**T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol**	16		S	AO-EXF	Exfiltration of data following collection of data on remote machine described in APT29_004 (this attack is now being used as a pivot to control access onto machines not directly on the internet)

[1] ESET_Operation_Ghost_Dukes.pdf (welivesecurity.com) [2019] {22}

[2] New sophisticated email-based attack from NOBELIUM - Microsoft Security Blog [2021] {18}

[3] eset_threat_report_t32021.pdf (welivesecurity.com) {33}

[4] CERTFR-2021-CTI-011.pdf (ssi.gouv.fr) (not directly ATT&CK but linked from [3])

[5] Assembling the Russian Nesting Doll: UNC2452 Merged into APT29 | Mandiant (additional to ATT&CK)

[6] PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs | Volexity (this is referenced on p7 of [1] in description of attack) [2016]

[7] CVE-2021-36934 - Security Update Guide - Microsoft - Windows Elevation of Privilege Vulnerability [2021] (via [3])

[8] APT29, IRON RITUAL, IRON HEMLOCK, NobleBaron, Dark Halo, StellarParticle, NOBELIUM, UNC2452, YTTRIUM, The Dukes, Cozy Bear, CozyDuke, Group G0016 | MITRE ATT&CK®

C.8 APT29_002

Dimension	***Technique***	***Notes***
Attribution	APT29	
Initial Access Vector	T1566.001 : Spearphishing Attachment	[1] The group's main initial tactic to breach a network is to send spearphishing emails that contain a link or an attachment Operation Ghost
Attack Origin	Russia	Via ATT&CK/TCERT
Target Location	US	[1] The group is primarily interested in spying on governments either in the West

		or in former USSR countries. Besides governments, the group has also targeted various organizations linked to NATO, think tanks, and political parties
Target Type	Ukraine based government department	[1] See above
Impact	Exfiltration (confidentiality)	
Vulnerabilities Exploited		
Related Attack Patterns		
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2018	[1]

	Sequence_ID	***APT29_002***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL	[1] The group's main initial tactic to breach a network is to send spearphishing emails that contain a link or an attachment
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN/NP-EXE	[1] See above PolyglotDuke dropped
3	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	2		S	IF-C2C	[1] PolyglotDuke this downloader shares several

							similarities with other samples from previous Dukes campaigns such as the use of Twitter to retrieve and decode its C&C server address, as well as a custom string encryption implementation
4	TA0005 : Defense Evasion	T1140 : Deobfuscate/Decode Files or Information	3		S	IF-DEV	[1] See above 'retrieve and decode its C&C server address'
5	TA0011 : Command & Control	T1071.001 Application Layer Protocol : Web Protocols	4		S	IF-C2C	[1] Fig 12 shows this as a three step initialisation with the C&C
6	TA0011 : Command & Control	T1071.001 Application Layer Protocol : Web Protocols	5		S	IF-C2C	[1] Fig 12 shows this as a three step initialisation with the C&C
7	TA0011 : Command & Control	T1105 : Ingress Tool Transfer	6		S	IF-C2C	[1] PolyglotDuke is a downloader that is used to download and drop the MiniDuke backdoor
8	TA0005 : Defense Evasion	T1140 : Deobfuscate/Decode Files or Information	7		S	IF-DEV	[1] PolyglotDoke dropper deobfuscated before running (Dropper desc p14)
9	TA0005 : Defense Evasion	T1218.011 : System Binary Proxy Execution: Rundll32	8		S	IF-DEV/NP-EXE	[1] dropper executed using rundll32.exe MiniDuke backdoor dropped
10	TA0005 : Defense Evasion	T1553.002 : <Subvert Trust Controls>:Code Signing	9		G	IF-DEV	[1] Invalid digital signature included in MiniDuke backdoor

11	**TA0005 : Defense Evasion**	**T1027 : Obfuscated Files or Information**	9		G	IF-DEV	[1] The backdoor is still written in pure x86 assembly but its size increased a lot – from 20 KB to 200+ KB. This is due to the addition of obfuscation, mainly control-flow flattening
12	**TA0003 : Persistence**	**T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**	9		S	IF-PER	[1] does not clarify persistence. E.g. from [2] used.
13	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	12		S	IF-C2C	[1] The network communication is relatively simple. It can use the GET, POST and PUT HTTP methods to contact the hardcoded C&C server.
14	**TA0007 : Discovery**	**T1082 : System Information Discovery**	13		S	NP-DIS	[1] Getting system information (hostname, ID, pipename, HTTP method)
15	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	14		S	IF-C2C	[1] The network communication is relatively simple. It can use the GET, POST and PUT HTTP methods to contact the hardcoded C&C server.
16	**TA0011 : Command & Control**	**T1105 : Ingress Tool Transfer**	15		S	IF-C2C	[1] During our investigation, we were not able to find a dropper for FatDuke. We believe the operators simply install the backdoor and establish persistence using

							the standard commands of an earlier stage backdoor.
17	TA0003 : Persistence	T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder	16		S	IF-PER	[1] We also noted that FatDuke generally replaced the second-stage binary, reusing the persistence mechanism already in place [1] The persistence we have seen is very standard. They use the registry key HKLM\SOFTWARE\ Microsoft\ Windows\CurrentVersion\Run and creatd a new value named Canon Gear and value C:\Program Files\Canon\Network ScanGear\Canocpc.exe. This launches the backdoor each time a user logs in.
18	TA0011 : Command & Control	T1071.001 Application Layer Protocol : Web Protocols	17		S	IF-C2C	[1] Example Fig 30 cycle Get HTML page (and extra image URL)
19	TA0011 : Command & Control	T1105 : Ingress Tool Transfer	18		S	IF-C2C	[1] Download PNG file
20	TA0005 : Defense Evasion	T1140 : Deobfuscate/Decode Files or Information	19		S	IF-DEV	[1] Decode and decrypt
21	TA0002 : Execution	T1059.003 : <Command and Scripting	20		S	NP-EXE	[1] Execute command

		Interpreter>:Windows Command Shell					
22	**TA0010 : Exfiltration**	**T1041 : Exfiltration Over C2 Channel**	21		S	AO-EXF	[1] See ATT&CK Techniques

[1] ESET_Operation_Ghost_Dukes.pdf (welivesecurity.com) [2019]

[2] FatDuke, Software S0512 | MITRE ATT&CK®

[3] MiniDuke, Software S0051 | MITRE ATT&CK®

C.9 APT29_003

Dimension	***Technique***	***Notes***
Attribution	APT29	
Initial Access Vector	T1566.001 : Spearphishing Attachment	[6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro
Attack Origin	Russia	Via ATT&CK/TCERT
Target Location	US	[6] The Dukes launched several waves of highly targeted spear phishing attacks against several U.S.-based think tanks and NGOs
Target Type	U.S.-based think tank	[6] See above
Impact	Exfiltration (confidentiality)	
Vulnerabilities Exploited		
Related Attack Patterns		

Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2016	[6] Wave 2

	Sequence_ID	***APT29_003***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL	[6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro
2	**TA0005 : Defense Evasion**	**T1497 : Virtualization/Sandbox Evasion**	1		G	IF-DEV	[6] The Macros contain several anti-VM checks designed to avoid executing in virtualized environments
3	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN/NP-EXE	[6] User executes HTML malicious Word Macro
4	**TA0005 : Defense Evasion**	**T1207 : Obfuscated Files or Information**	3		G	IF-DEV	[6] Alternate data stream (ADS) PNG file with the PowerDuke backdoor component hidden and encrypted within using Tiny Encryption Algorithm (TEA). (NTFS Alternate Data Streams: The Good and the Bad (foldersecurityviewer.com))
5	**TA0003 : Persistence**	**T1547.001 : <Boot or Logon Autostart**	3		S	IF-PER	[6] PowerDuke backdoor file dropped to

		Execution>:Registry Run Keys / Startup Folder					"%APPDATA\Roaming\HP\" with persistence via HKCU Run Key "ToolboxFX" (rundll32.exe %APPDATA\Roaming\ HP\fywhx.dll #2). Connects directly to 185.132.124.43:443 for command and control.
6	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	5		S	IF-C2C	[6] See above PowerDuke Backdoor begins communication Limited info on C&C approach but from [7] The malware attempts to blend in with normal network traffic as much as possible. This is done with a handful of different tactics. Communication is often done with the HTTP protocol. Some of the malware will attempt to use realistic looking User-Agent strings with the requests
7	**TA0007 Discovery**	**T1083 : File and Directory Discovery**	6		S	NP-DIS	Specific of attack sequence is not made clear so a simple example is constructed of an Exfiltration attack is included here. The choice of exfiltration is based on APT characterisation in [1]. The actions noted are based on the Backdoor capability documented below

							(from [6]) and MITRE ATT&CK documented Technique use.
8	**TA0007 Discovery**	**T1083 : File and Directory Discovery**	7		S	NP-DIS	Find information about files visible from backdoor
9	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	8		S	IF-C2C	Send info back Then next command. For these couples these will be conflated as here.
10	**TA0010 : Exfiltration**	**T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol**	9		S	AO-EXF	[6] fgetp
11	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	10		S	IF-C2C	Then next command.
12	**TA0002 : Execution**	**T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**	11		S	NP-EXE	[6] Run start a process via CreateProcessW # runs cmd.exe /c and gets the output via Named Pipe and sends the data back
13	**TA0010 : Exfiltration**	**T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol**	12		S	AO-EXF	See above

14	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	13		S	IF-C2C	Then next command.
15	**TA0009 : Collection**	**T1113 : Screen Capture**	14		S	AO-COL	wnd gets the text of the current foreground window
16	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol : Web Protocols**	15		S	IF-C2C	Then next command (and return the screen capture)

[1] ESET_Operation_Ghost_Dukes.pdf (welivesecurity.com) [2019]

[2] New sophisticated email-based attack from NOBELIUM - Microsoft Security Blog [2021]

[3] eset_threat_report_t32021.pdf (welivesecurity.com)

[4] CERTFR-2021-CTI-011.pdf (ssi.gouv.fr) (not directly ATT&CK but linked from [3])

[5] Assembling the Russian Nesting Doll: UNC2452 Merged into APT29 | Mandiant (additional to ATT&CK)

[6] PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs | Volexity (this is referenced on p7 of [1] in description of attack) [2016]

[7] The Dukes of Moscow - VMware Security Blog - VMware [2020]

C.10 APT29_004

Dimension	***Technique***	***Notes***
Attribution	APT29	
Initial Access Vector	T1566.001 : Spearphishing Attachment	[6] The second attack wave that Volexity observed leveraged a Microsoft Word document

		with a malicious embedded macro
Attack Origin	Russia	Via ATT&CK/TCERT
Target Location	US	[6] The Dukes launched several waves of highly targeted spear phishing attacks against several U.S.-based think tanks and NGOs
Target Type	U.S.-based think tank	[6] See above
Impact	Exfiltration (confidentiality)	
Vulnerabilities Exploited	CVE-2021-36934	
Related Attack Patterns		
Preceded By	APT29_001	This is a very brief example to demonstrate codification of attacks pivoting through initial access platform. APT29_001 will receive the data collected and exfiltrate.
Schema Version	0.1	To allow for future new sequence models
Date	2016	[6] Wave 2

	Sequence_ID	***APT29_004***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***

1	**TA0008 : Lateral Movement**	**T1021.002 : <Remote Services>:SMB/Windows Admin Shares**	0		S	NP-LMV	[3] Via Sharp-SMBExec
2	**TA0008 : Lateral Movement**	**T1570 : Lateral Tool Transfer**	1		S	NP-LMV	[3] They also usually deploy additional Cobalt Strike loaders (DLL- or PowerShell-based) that would load SMB beacons. These can be used to control machines in the same network that are not directly connected to the internet
3	**TA0009 : Collection**	**T1213 : Data from Information Repositories**	2		S	AO-COL	[3] Attacker able to collect data via this account. Here using the Cobalt Strike installation above

[1] ESET_Operation_Ghost_Dukes.pdf (welivesecurity.com) [2019] {22}

[2] New sophisticated email-based attack from NOBELIUM - Microsoft Security Blog [2021] {18}

[3] eset_threat_report_t32021.pdf (welivesecurity.com) {33}

[4] CERTFR-2021-CTI-011.pdf (ssi.gouv.fr) (not directly ATT&CK but linked from [3])

[5] Assembling the Russian Nesting Doll: UNC2452 Merged into APT29 | Mandiant (additional to ATT&CK)

[6] PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs | Volexity (this is referenced on p7 of [1] in description of attack) [2016]

[7] CVE-2021-36934 - Security Update Guide - Microsoft - Windows Elevation of Privilege Vulnerability (via [3])

[8] They See Me Roaming: Following APT29 by Taking a Deeper Look at Windows Credential Roaming | Mandiant (via Google search "APT29 Lateral Movement" and APT29 Persistence / Lateral Movement via Windows Credential Roaming | Threat SnapShot - YouTube)

C.11 APT32_001

Dimension	***Technique***	***Notes***
Attribution	APT32	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	Vietnam	
Target Location		TBC
Target Type		
Impact	Exfiltration (confidentiality)	[3] Cybereason concluded the main motivation behind the attack was cyber espionage
Vulnerabilities Exploited		
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2017	

	Sequence_ID	*APT32_001*		*Ver*	*0.1*		
ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*
1	TA0001 : Initial Access	T1566.001 : Spearphishing Attachment	0		S	IF-DEL/IF-SEN	[3] Two types of payloads were found in the spear-phishing emails:
2	TA0002 : Execution	T1204.002 : User Execution : Malicious File	1		S	IF-SEN	[3] Word File with malicious macro delivering Cobalt Strike Beacon
3	TA0003 : Persistence	T1053.005 : <Scheduled Task/Job>:Scheduled Task,	2		S	IF-PER	[3] The malicious macro creates two scheduled tasks that download files camouflaged as ".jpg" files from the C&C server
4	TA0011 : Command & Control	T1105 : Ingress Tool Transfer	3		S	IF-C2C	See above
5	TA0005 : Defense Evasion	T1036.005 : <Masquerading>:Match Legitimate Name or Location	4		G	IF-DEV	See above
6	TA0005 : Defense Evasion	T1218.005 System Binary Proxy Execution: Mshta	4		S	IF-DEV/NP-EXE	[3] The purpose of the scheduled task is to download another payload from the C&C

							server: schtasks /create /sc MINUTE /tn "Windows Error Reporting" /tr "mshta.exe () about:'<script
7	TA0011 : Command & Control	T1105 : Ingress Tool Transfer	6		S	IF-C2C	See above
8	TA0002 : Execution	T1059.005 : <Command and Scripting Interpreter>:Visual Basic	7		S	NP-EXE	[3] The content of the “microsoftp.jpg” is a script that combines vbscript and PowerShell
9	TA0002 : Execution	T1059.001 : <Command and Scripting Interpreter>:PowerShell	8		S	NP-EXE	As above
10	TA0005 : Defense Evasion	T1027 : Obfuscated Files or Information	9		G	IF-DEV	[3] Obfuscated PowerShell delivering Cobalt Strike Beacon
11	TA0002 : Execution	T1059.001 : <Command and Scripting Interpreter>:PowerShell	9		S	NP-EXE	As above
12	TA0005 : Defense Evasion	T1027 : Obfuscated Files or Information	11		G	IF-DEV	[3] Obfuscated PowerShell delivering Cobalt Strike Beacon

13	**TA0011 : Command & Control**	**T1105 : Ingress Tool Transfer**	11		S	IF-C2C	[3] The PowerShell process will then download the new 'image.jpg' payload, which is actually another obfuscated PowerShell payload:
14	**TA0003 : Persistence**	**T1574 Hijack Execution Flow**	13		G	IF-PER	[3] Backdoor exploits DLL hijacking against Wsearch Service [3] The attackers exploited a DLL hijacking vulnerability in a legitimate Google Update binary, which was deployed along with a malicious DLL (goopdate.dll).
15	**TA0003 : Persistence**	**T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder**	13		S	IF-PER	[3] The attackers used a malicious Outlook backdoor macro to communicate with the C2 servers and exfiltrate data. To make sure the malicious macro ran, they edited a specific registry value to create persistence
16	**TA0003 : Persistence**	**T1543.003 : <Create or Modify System**	15		S	IF-PER	[3] The attackers created and/or modified Windows

		Process>:Windows Service					Services to ensure the loading of the PowerShell scripts on the compromised machines
17	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	16		S	IF-C2C	Cobalt Strike Fileless Infrastructure (HTTP)
18	**TA0007 Discovery**	**T1083 File and Directory Discovery**	17		S	NP-DIS	
19	**TA0007 Discovery**	**T1082 System Information Discovery**	18		S	NP-DIS	
20	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	19		S	IF-C2C	Cobalt Strike Fileless Infrastructure (HTTP)
21	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	20		S	IF-C2C	Cobalt Strike Fileless Infrastructure (HTTP)
22	**TA0009 : Collection**	**T1056.001 : <Input Capture>:Keylogging**	21		S	AO-COL	
23	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	22		S	IF-C2C	Cobalt Strike Fileless Infrastructure (HTTP)
24	**TA0005 : Defense Evasion**	**T1218.011 : <System Binary Proxy Execution>:Rundll32**	23		S	IF-DEV/NP-EXE	Added as an example. It is not entirely clear when it is used but notes that it is used to download additional payloads e.g. COM scriptlets

							[3] The attackers downloaded COM scriptlets using regsvr32.exe
25	**TA0002 : Execution**	**T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**	24		S	NP-EXE	See above

[1] OceanLotus ships new backdoor using old tricks | WeLiveSecurity [2018]

[2] Operation Cobalt Kitty: A large-scale APT in Asia carried out by the OceanLotus Group (cybereason.com) [2017]

[3] Cybereason Labs Analysis Operation Cobalt Kitty.pdf (hubspot.net) – Attack Lifecycle [2018]

[4] Fake or Fake: Keeping up with OceanLotus decoys | WeLiveSecurity [2019]

[5] Vietnamese Threat Actors APT32 Targeting Wuhan Government and Chinese Ministry of Emergency Management in Latest Example of COVID-19 Related Espionage | Mandiant [2020]

[6] Click-and-Bait_Vietnamese-Human-Rights-Defenders-Targeted-with-Spyware-Attacks.pdf (amnestyusa.org) [2021]

C.12 Lazarus_Group_001

Dimension	***Description***	***Notes***
Attribution	Lazarus Group	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	North Korea	
Target Location		TBC
Target Type		TBC
Impact		TBC

Vulnerabilities Exploited	CVE-2018-4878	
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2018	

	Sequence_ID	***Lazarus_Group_001***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	[1] Based on our analysis, financial organizations in Turkey were targeted via spear phishing emails
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	[1] Containing a malicious Microsoft Word document
3	**TA0002 : Execution**	**T1203 : Exploitation for Client Execution**	2		S	IF-EXP	[1] The document contains an embedded Flash script that exploits CVE-2018-4878 and downloads and executes the DLL

							implant from falcancoin.io
4	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	3		G		See above
5	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	3		S	IF-C2C	See above
6	**TA0005 Defense Evasion**	**T1036 Masquerading**	5		G		[1] The implants (DLLs) are disguised as ZIP files
7	**TA0005 Defense Evasion**	**T1055.001 Process Injection: Dynamic-link Library Injection**	5		S	IF-DEV	[1] To mask itself, it can run as a regular library loaded into a legitimate process Pivot happens here
8	**TA0003 : Persistence**	**T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**					Replaced by 6 above after further investigation
8	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	7		S	IF-C2C	[1] The malware initiates communication with the control server by sending it an HTTP POST request with

							additional optional HTTP data
9	**TA0007 Discovery**	**T1083 File and Directory Discovery**	8		S	NP-DIS	[1] Recursively generate a list of files in a directory and send to the control server
10	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	9		G	IF-C2C	[1] After every action is performed the malware sends a response to the control server indicating whether the action was successful
11	**TA0011 : Command & Control**	**T1041 : Exfiltration Over C2 Channel**	10		S	IF-C2C	[1] Read a specified file's contents and send the data to the control server
12	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	11		S	IF-C2C	Next command in from control in this case as below
13	**TA0011 : Command & Control**	**T1041 : Exfiltration Over C2 Channel**	12		S	AO-EXF	Access Begins [1] Read a specified file's

							contents and send the data to the control server
14	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	13		S	IF-C2C	Next command in from control in this case as below
15	**TA0002 : Execution**	**T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**	14		S	NP-EXE	This is an example. Execution will have a goal assume further discovery
16	**TA0007 Discovery**	**T1087.002 : <Account Discovery>:Domain Account**	15		S	NP-DIS	This is an example. Execution will have a goal assume further discovery
17	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	16		S	IF-C2C	After every action is performed the malware sends a response to the control server indicating whether the action was successful

[1] Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant | McAfee Blog [2018]

[2] Lazarus targets defense industry with ThreatNeedle | Securelist [2021]

[3] Operation (노스 스타) North Star A Job Offer That's Too Good to be True? | McAfee Blog [2020]

[4] North Korea's Lazarus APT leverages Windows Update client, GitHub in latest campaign (malwarebytes.com) [2022]

[5] LolZarus: Lazarus Group Incorporating Lolbins into Campaigns | Qualys Security Blog [2022]

C.13 Lazarus_Group_002

Dimension	***Description***	***Notes***
Attribution	Lazarus Group	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	North Korea	
Target Location		TBC
Target Type		TBC
Impact		TBC
Vulnerabilities Exploited	CVE-2018-4878	
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2018	

	Sequence_ID	***Lazarus_Group_002***		***Ver***	***0.1***		

ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	[1] Based on our analysis, financial organizations in Turkey were targeted via spear phishing emails
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	[1] Containing a malicious Microsoft Word document
3	**TA0002 : Execution**	**T1203 : Exploitation for Client Execution**	2		S	IF-EXP	[1] The document contains an embedded Flash script that exploits CVE-2018-4878 and downloads and executes the DLL implant from falcancoin.io
4	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	3		G	IF-C2C	See above
5	**TA0011 : Command & Control**	**T1105 Ingress Tool Transfer**	3		S	IF-C2C	See above

6	TA0005 Defense Evasion	T1036 Masquerading	5		G	IF-DEV	[1] The implants (DLLs) are disguised as ZIP files
7	TA0005 Defense Evasion	T1055.001 Process Injection: Dynamic-link Library Injection	5		S	IF-DEV	[1] To mask itself, it can run as a regular library loaded into a legitimate process
8	TA0003 : Persistence	T1574.002 : <Hijack Execution Flow>:DLL Side-Loading					Replaced by 6 above after further investigation
8	TA0011 : Command & Control	T1071.001 : <Application Layer Protocol>:Web Protocols	7		S	IF-C2C	[1] The malware initiates communication with the control server by sending it an HTTP POST request with additional optional HTTP data
9	TA0007 Discovery	T1083 : File and Directory Discovery	8		S	NP-DIS	[1] Recursively generate a list of files in a directory and send to the control server

10	**TA0007 Discovery**	**T1082 : System Information Discovery**	9		S	NP-DIS	[1] Send information for all drives
11	**TA0007 Discovery**	**T1016 : System Network Configuration Discovery**	10		S	NP-DIS	[1] Gather network addresses
12	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	11		G	IF-C2C	[1] After every action is performed the malware sends a response to the control server indicating whether the action was successful
13	**TA0011 : Command & Control**	**T1041 : Exfiltration Over C2 Channel**	11		S	IF-C2C/AO-EXF	[1] Read a specified file's contents and send the data to the control server
14	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	13		S	IF-C2C	Next command in from control in this case as below
15	**TA0011 : Command & Control**	**T1041 : Exfiltration Over C2 Channel**	14		S	IF-C2C/AO-EXF	[1] Read a specified file's contents and send the data to the control server

16	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	15		S	IF-C2C	Next command in from control in this case as below
17	**TA0002 : Execution**	**T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**	16		S	NP-EXE	
18	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	17		S	IF-C2C	After every action is performed the malware sends a response to the control server indicating whether the action was successful

[1] Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant | McAfee Blog [2018]

[2] Lazarus targets defense industry with ThreatNeedle | Securelist [2021]

[3] Operation (노스 스타) North Star A Job Offer That's Too Good to be True? | McAfee Blog [2020]

[4] North Korea's Lazarus APT leverages Windows Update client, GitHub in latest campaign (malwarebytes.com) [2022]

[5] LolZarus: Lazarus Group Incorporating Lolbins into Campaigns | Qualys Security Blog [2022]

[6] Lazarus targets defense industry with ThreatNeedle | Securelist [2021] – For T1049 but save for a new attack

C.14 MuddyWater_001

Dimension	*Description*	*Notes*
Attribution	MuddyWater	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	Iran	
Target Location	Turkey, Pakistan, Tajikistan [2] Middle East [1]	
Target Type	[2] Government, Defense	
Impact		
Vulnerabilities Exploited		
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2018	[2] From January 2018 to March 2018, we observed attackers leveraging the latest code execution and persistence techniques …

	Sequence_ID	*MuddyWater_001*		*Ver*	*0.1*		
ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*

1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	[2] The spear phishing emails and attached malicious macro documents typically have geopolitical themes
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	[2] The first part of the campaign (From Jan. 23, 2018, to Feb. 26, 2018) used a macro-based document that dropped a VBS file and an INI file
3	**TA0005 : Defense Evasion**	**T1207 Obfuscated Files or Information**	2		G	IF-DEV	[2] The INI file contains the Base64 encoded PowerShell command ….
4	**TA0003 : Persistence**	**T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**	2		S	IF-PER	Assuming a similar approach to second approach [2] After dropping the three files, the macro will set the following registry key to achieve persistence
5	**TA0002 : Execution**	**T1059.005 : <Command and Scripting Interpreter>:Visual Basic**	4		S	NP-EXE	[2] …. (from 3) which will be decoded and executed by PowerShell using the command line generated by the VBS file on execution using WScript.exe

6	**TA0002 : Execution**	**T1218.005 : <System Binary Proxy Execution>:Mshta**	5		S	NP-EXE	[2] One such example of the VBS invoking PowerShell via MSHTA is shown
7	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	6		S	IF-DEV	[2] The main function performed by the SCT file is to Base64 decode the contents of WindowsDefender.ini file and execute the decoded PowerShell Script
8	**TA0002 : Execution**	**T1059.001 : <Command and Scripting Interpreter>:PowerShell**	7		S	NP-EXE	See above
9	**TA0011 : Command & Control**	**T1132.001 : <Data Encoding>:Standard Encoding**	8		G	IF-C2C	[2] Two approaches to message encoding provided in text below
10	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	8		S	IF-C2C	
11	**TA007 Discovery**	**T1083 File and Directory Discovery**	10		S	NP-DIS	Discovery examples
12	**TA007 Discovery**	**T1082 System Information Discovery**	11		S	NP-DIS	Discovery examples
13	**TA007 Discovery**	**T1016 System Network Configuration Discovery**	12		S	NP-DIS	Discovery examples
14	**TA007 Discovery**	**T1049 System Network Connections Discovery**	13		S	NP-DIS	Discovery examples

15	**TA0011 : Command & Control**	**T1132.001 : <Data Encoding>:Standard Encoding**	14		G	IF-C2C	C2 Exfiltration example
16	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	14		S	IF-C2C/AO-EXF	

[1] Muddying the Water: Targeted Attacks in the Middle East (paloaltonetworks.com) [2017]

[2] Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign | Mandiant [2018]

[3] MuddyWater expands operations | Securelist [2018]

[4] Probable Iranian Cyber Actors, Static Kitten, Conducting Cyberespionage Campaign Targeting UAE and Kuwait Government Agencies (anomali.com) [2021]

[5] Earth Vetala MuddyWater Continues to Target Organizations in the Middle East (trendmicro.com) [2021]

C.15 MuddyWater_002

Dimension	***Description***	***Notes***
Attribution	MuddyWater	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	Iran	
Target Location	Turkey, Pakistan, Tajikistan [2] Middle East [1]	
Target Type	[2] Government, Defense	
Impact		

Vulnerabilities Exploited		
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2018	[2] From January 2018 to March 2018, we observed attackers leveraging the latest code execution and persistence techniques …

	Sequence_ID	***MuddyWater_002***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***
1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	[2] The spear phishing emails and attached malicious macro documents typically have geopolitical themes
2	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	[2] The first part of the campaign (From Jan. 23, 2018, to Feb. 26, 2018) used a macro-based document that dropped a VBS file and an INI file

3	**TA0005 : Defense Evasion**	**T1207 Obfuscated Files or Information**	2		G	IF-DEV	[2] The INI file contains the Base64 encoded PowerShell command ….
4	**TA0003 : Persistence**	**T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**	2		S	IF-PER	[2] After dropping the three files, the macro will set the following registry key to achieve persistence
5	**TA0005 : Defense Evasion**	**T1218.003 : System Binary Proxy Execution: CMSTP**	4		S	NP-EXE	[2] Upon system restart, cmstp.exe will be used to execute the SCT file indirectly through the INF file.
6	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	5		S	IF-DEV	[2] The main function performed by the SCT file is to Base64 decode the contents of WindowsDefender.ini file ….
7	**TA0002 : Execution**	**T1059.001 : <Command and Scripting Interpreter>:PowerShell**	6		S	NP-EXE	[2] … and execute the decoded PowerShell Script
8	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	7		S	IF-DEV	[2] The PowerShell script employs several layers of obfuscation to hide its actual functionality. The biggest section of the

							PowerShell script is XOR encoded using a single byte key
9	**TA0011 : Command & Control**	**T1573.002 : Encrypted Channel: Asymmetric Cryptography**	8		G	IF-C2C	[2] The second section of the PowerShell script has the ability to perform encryption and decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA
10	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	8		S	IF-C2C	
11	**TA007 Discovery**	**T1083 File and Directory Discovery**	10		S	NP-DIS	Discovery examples from known behaviours
12	**TA007 Discovery**	**T1082 System Information Discovery**	11		S	NP-DIS	See [2] powershell example above
13	**TA007 Discovery**	**T1016 System Network Configuration Discovery**	12		S	NP-DIS	See [2] powershell example above
14	**TA007 Discovery**	**T1049 System Network Connections Discovery**	13		S	NP-DIS	See [2] powershell example above
15	**TA0011 : Command & Control**	**T1573.002 : Encrypted Channel: Asymmetric Cryptography**	14		G	IF-C2C	[2] The second section of the PowerShell script has the ability to perform encryption and

							decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA
16	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	14		S	IF-C2C	Return data And send message (conflated for brevity)
17	**TA0002 : Execution**	**T1559.001 : <Inter-Process Communication>:Component Object Model**	16		S	NP-EXE	[2] Outlook - Leverage Outlook.Application COM object for code execution
18	**TA0011 : Command & Control**	**T1573.002 : Encrypted Channel: Asymmetric Cryptography**	17		G	IF-C2C	[2] The second section of the PowerShell script has the ability to perform encryption and decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA
19	**TA0011 : Command & Control**	**T1071.001 : <Application Layer Protocol>:Web Protocols**	17		S	IF-C2C	Return data

C.16 Mustang_Panda_001

Dimension	*Description*	*Notes*
Attribution	Mustang Panda	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	China	
Target Location	Hong Kong	
Target Type	Catholic Church	
Impact	Exfiltration	[1] “making his successor a valuable target for intelligence gathering ahead of the deal’s expiry and likely renewal in September 2020”
Vulnerabilities Exploited		
Related Attack Patterns		
Preceded By	NA	
Schema Version	0.1	
Date	2020	

	Sequence_ID	*Mustang_Panda_001*		*Ver*	*0.1*		
ID	*Tactic*	*Technique*	*Pred*	*TInc*	*S/G*	*KC Step*	*Notes*

1	**TA0001 : Initial Access**	**T1566.001 : Spearphishing Attachment**	0		S	IF-DEL/IF-SEN	[1] The lure document … was used to deliver a customized PlugX payload …. Given that the letter was directly addressed to this individual, it is likely that he was the target of a spearphishing attempt
2	**TA0005 : Defense Evasion**	**T1036 Masquerading**	1		G	IF-DEV	[1] RedDelta used ZIP files containing legitimate executables masquerading as lure documents
3	**TA0002 : Execution**	**T1204.002 : User Execution : Malicious File**	1		S	IF-SEN	In [1] it is not explicitly stated but assuming that target opens zip and lure 'documents'. See below
4	**TA0005 : Defense Evasion**	**T1574.002 Hijack Execution Flow: DLL Side-Loading**	3		S	IF-DEV/NP-EXE	[1] This legitimate executable is used to load a malicious DLL also present within the ZIP file through DLL sideloading, before the target is shown a decoy document
5	**TA0011 : Command and Control**	**T1105 : Ingress Tool Transfer**	4		S	IF-C2C	[1] Wwlib.dll initializes the loading stage by downloading, decoding, and executing an XOR-

							encoded Windows executable file, hk.dat
6	TA0005 : Defense Evasion	T1027 : Obfuscated Files or Information	5		G	IF-DEV	As above
7	TA0005 : Defense Evasion	T1140 : Deobfuscate/Decode Files or Information	5		S	IF-DEV	As above
8	TA0002 : Execution	T1059.003 : Command and Scripting Interpreter: Windows Command Shell	7		S	NP-EXE	As above. The execution method is not clear so this is a holding suggestion (from known Tech use of this APT) See also [2] for additional suggestions
9	TA0005 : Defense Evasion	T1140 : Deobfuscate/Decode Files or Information	8		S	IF-DEV	As above
10	TA0002 : Execution	T1059.003 : Command and Scripting Interpreter: Windows Command Shell	9		S	NP-EXE	See also above [1] Next, “hk.exe” is executed and creates copies of the files “adobeupdate.dat,” “hex.dll,” and itself renamed as “AAM Updates.exe” in the folder “C:\ProgramData\AAM UpdatesIIw.”
11	TA0002 : Execution	T1059.003 : Command and Scripting	10		S	NP-EXE	See also above

		Interpreter: Windows Command Shell					[1] "AAM Updates.exe" is then executed, starting the installation process by sideloading the malicious "hex.dll." "Hex.dll" will decode and execute "adobeupdate.dat," which ultimately leads to the execution of the RedDelta PlugX variant in memory
12	**TA0005 : Defense Evasion**	**T1574.002 Hijack Execution Flow: DLL Side-Loading**	11		S	IF-DEV/NP-EXE	See above
13	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	12		S	IF-DEV	As above
14	**TA0002 : Execution**	**T1059.003 : Command and Scripting Interpreter: Windows Command Shell**	13		S	NP-EXE	See also above [1] …… which ultimately leads to the execution of the RedDelta PlugX variant in memory We are not told how so this is a dummy technique
15	**TA0011 : Command & Control**	**T1573.001 Encrypted Channel: Symmetric Cryptography**	14		G	IF-C2C	[1] RedDelta uses RC4 encryption Out message/In message

16	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol: Web Protocols**	14		S	IF-C2C	[1] RedDelta uses HTTP
17	**TA0007 : Discovery**	**T1049 : System Network Connections Discovery**	16		S	NP-DIS	[5] Report has no info on action taken so a simple example
18	**TA0007 : Discovery**	**T1082 : System Information Discovery**	17		S	NP-DIS	See above
19	**TA0007 : Discovery**	**T1082 : System Information Discovery**	18		S	NP-DIS	See above
20	**TA0011 : Command & Control**	**T1573.001 Encrypted Channel: Symmetric Cryptography**	19		G	IF-C2C	[1] RedDelta uses RC4 encryption
21	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol: Web Protocols**	19		S	IF-C2C	[1] RedDelta uses HTTP
22	**TA0009 : Collection**	**T1113 Screen Capture**	21		S	AO-COL	[5]
23	**TA0011 : Command & Control**	**T1573.001 Encrypted Channel: Symmetric Cryptography**	22		G	IF-C2C	[1] RedDelta uses RC4 encryption
24	**TA0011 : Command & Control**	**T1071.001 Application Layer Protocol: Web Protocols**	22		S	IF-C2C	[1] RedDelta uses HTTP

[1] Chinese State-Sponsored Group 'RedDelta' Targets the Vatican and Catholic Organizations (recordedfuture.com) [2020]

[2] TA416 Goes to Ground and Returns with a Golang PlugX Malware Loader | Proofpoint US [2020]

[3] An update on the threat landscape (blog.google) [2022]

[4] Chinese APT Bronze President Mounts Spy Campaign on Russian Military (darkreading.com) [2022] (via malpedia)

[5] PlugX (Malware Family) (fraunhofer.de) via malpedia

C.17 Sandworm_001

Dimension	***Description***	***Notes***
Attribution	Sandworm Team	
Initial Access Vector	T1566.001 : Spearphishing Attachment	[2]
Attack Origin	Russia	
Target Location	Ukraine	[2]
Target Type	Ukrainian government organization	[2]
Impact		[2] cyber-espionage
Vulnerabilities Exploited	CVE-2014-4114	
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2014	

	Sequence_ID	***Sandworm_001***		***Ver***	***0.1***		

ID	Tactic	Technique	Pred	TInc	S/G	KC Step	Notes
1	TA0001 : Initial Access	T1566.001 : Spearphishing Attachment	0		S	IF-DEL/IF-SEN	[2] On September 3rd, our research and labs teams discovered that the spear-phishing attacks
2	TA0002 : Execution	T1204.002 : User Execution : Malicious File	1		S	IF-SEN	[2] A weaponized PowerPoint document was observed in these attacks
3	TA0002 : Execution	T1203 : Exploitation for Client Execution	2		S	IF-EXP	[2] The user tricked into execution (CVE-2014-4114 allow remote attackers to execute arbitrary code. Via crafted OLE object in office file
4	TA0011 : Command and Control	T1105 : Ingress Tool Transfer	3		S	IF-C2C	[6] An attacker who successfully exploited this vulnerability could assume the role of the current user on the target machine [7] In the case of the live sample exploit PPSX file I examined, it automatically downloaded the payload from a remote SMB share
5	TA0003 : Persistence	T1547 : Boot or Logon Autostart Execution	4		S	IF-PER	[7] INF contains “HKLM,Software\\ Microsoft\\Windows\\

							CurrentVersion\\ RunOnce,Install" [1] [2] BlackEnergy installed
6	**TA0006 : Credential Access**	**T1056.001 Input Capture: Keylogging**	5		S	NP-CAC	[1] used a particular variant of malware called BlackEnergy to steal user credentials
7	**TA0010 : Exfiltration**	**T1041 : Exfiltration Over C2 Channel**	6		G	AO-EXF	[1] credentials must have been exfiltrated in some form as stated that they are used to access the SCADA networks
8	**TA0011 : Command and Control**	**T1071.001 : Application Layer Protocol: Web Protocols**	6		S	IF-C2C	[9] Noted in ATT&CK BlackEnergy description
9	**TA0007 : Discovery**	**T1082 System Information Discovery**	8		S	NP-DIS	[8] [9] Noted in ATT&CK BlackEnergy description
10	**TA0010 : Exfiltration**	**T1041 : Exfiltration Over C2 Channel**	9		G	AO-EXF	[1] System info exfiltrated
11	**TA0011 : Command and Control**	**T1071.001 : Application Layer Protocol: Web Protocols**	9		S	IF-C2C	[9] Noted in ATT&CK BlackEnergy description
12	**TA0007 : Discovery**	**T1046 Network Service Discovery**	11		S	NP-DIS	[9] Noted in ATT&CK BlackEnergy description

13	**TA0010 : Exfiltration**	**T1041 : Exfiltration Over C2 Channel**	12		G	AO-EXF	[1] System info exfiltrated
14	**TA0011 : Command and Control**	**T1071.001 : Application Layer Protocol: Web Protocols**	12		S	IF-C2C	[9] Noted in ATT&CK BlackEnergy description
15	**TA0011 : Command and Control**	**T1105 : Ingress Tool Transfer**	14		S	IF-C2C	[1] Sandworm used destructive software Killdisk
16	**TA0040 : Impact**	**T1485 Data Destruction**	15		S	AO-TMA	[1] Delete computer event logs and other files
17	**TA0040 : Impact**	**T1529 System Shutdown/Reboot**	16		S	AO-TMA	[1] Reboot infected computers

[1] Six Russian GRU Officers Charged in Connection with Worldwide Deployment of Destructive Malware and Other Disruptive Actions in Cyberspace: Unsealed Indictment (justice.gov) [2020]

[2] Sandworm Zero Day Vulnerability | iSIGHT Partners (archive.org) [2014]

[3] Overview of the Cyber Weapons Used in the Ukraine - Russia War | Trustwave [2022] (via Malpedia)

[4] Russian State-Sponsored and Criminal Cyber Threats to Critical Infrastructure | CISA [2022] (via Malpedia)

[5] CVE - CVE-2014-4114 (mitre.org) (via Google) [2014]

[6] Windows OLE RCE - The Sandworm Exploit (controlcase.com) [2014] (via Google)

[7] Windows OLE RCE Exploit MS14-060 (CVE-2014-4114) - Sandworm - Security SiftSecurity Sift [2014] (via Google)

[8] Russian malware used by 'privateer' hackers against Ukrainian government | Technology | The Guardian (archive.org) [2015]

[9] BlackEnergy, Software S0089 | MITRE ATT&CK® [2017]

C.18 Tropic_Trooper_001

Dimension	***Description***	***Notes***
Attribution	Tropic Trooper	
Initial Access Vector	T1566.001 : Spearphishing Attachment	
Attack Origin	China	
Target Location	Taiwan	
Target Type	Government	
Impact	Exfiltration (confidentiality)	
Vulnerabilities Exploited	CVE-2012-0158	
Related Attack Patterns		TBC
Preceded By	NA	Reference Sequence_ID
Schema Version	0.1	To allow for future new sequence models
Date	2015	

	Sequence_ID	***Tropic_Trooper_001***		***Ver***	***0.1***		
ID	***Tactic***	***Technique***	***Pred***	***TInc***	***S/G***	***KC Step***	***Notes***

1	TA0001 : Initial Access	T1566.001 : Spearphishing Attachment	0		S	IF-DEL/IF-SEN	[1] The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158
2	TA0005 : Defense Evasion	T1036 : Masquerading	1		G	IF-DEV	[1] The delivery document uses the XLSX extension typically used by OpenXML documents, but the file itself is actually an OLE (XLS) document
3	TA0005 : Defense Evasion	T1027 : Obfuscated Files or Information	1		G	IF-DEV	… which stores XLSX ciphertext and the information needed for decryption in an OLE document
4	TA0002 : Execution	T1204.002 : User Execution : Malicious File	1		S	IF-SEN	The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158
5	TA0002 : Execution	T1203 : Exploitation for Client Execution	4		S	IF-EXP	The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158 ([8] allows remote attackers to execute arbitrary code via a crafted (a) web site, (b) Office document, or (c) .rtf file that triggers "system state" corruption, as exploited in the wild in April 2012, aka "MSCOMCTL.OCX RCE Vulnerability.")
6	TA0002 : Execution	T1059 Command and Scripting Interpreter	5		S	NP-EXE	[1] First shell code in malicious document [1] The embedded shellcode enumerates open handles for a file with a size greater than 0xa6f0

7	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	6		S	IF-DEV	[1] The shellcode then decrypts the first 0xc0 (decimal 192) DWORDs of the data read from the file using an XOR algorithm
8	**TA0002 : Execution**	**T1059 Command and Scripting Interpreter**	7		S	NP-EXE	[1] Second shell code in malicious document
9	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	8		S	IF-DEV	[1 The secondary shellcode starts by resolving the following API functions using a ROT13 hashing algorithm
10	**TA0003 : Persistence**	**T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**	9		S	IF-PER	[1] The shellcode then creates a string that it uses to create a registry key to automatically run the final payload each time the system starts
11	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	10		S	IF-DEV	[1] The shellcode then enters a decryption loop to convert the embedded payload from ciphertext to cleartext. [1] the algorithm decrypts what is an embedded portable executable that acts as the payload in this attack. The embedded payload is written to %APPDATA\Identities\Identities.ocx [1] Delivered Payload – Poison Ivy
12	**TA0007 : Discovery**	**T1518.001 : Software Discovery: Security Software Discovery**	11		S	NP-DIS	[1] Before running the above command to open the decoy document, the shellcode enumerates the running processes on the system, specifically looking

							for processes created for an executable with a filename that starts with “avp.”, presumably in an attempt to find Kaspersky’s antivirus process. If the process is found, the shellcode will not open the decoy document and exits.
13	**TA0003 : Persistence**	**T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**	12		S	IF-PER	[1] The shellcode then creates a string that it uses to create a registry key to automatically run the final payload each time the system starts [1] When the system starts up, the persistence registry key will launch the Identities.ocx payload and call its “SSSS” exported function. The “SSSS” function checks to make sure that the DLL is running within the context of a “rundll32.exe” process …
14	**TA0005 : Defense Evasion**	**T1140 : Deobfuscate/Decode Files or Information**	13		S	IF-DEV	[1] … and then begins piecing 0x141B bytes of data together in the correct order to build the shellcode of the Poison Ivy Trojan
15	**TA0011 : Command and Control**	**T1573.001 : Encrypted Channel: Symmetric Cryptography**	14		S	IF-C2C	[9] Example initial communication
16	**TA0009 : Collection**	**T1005 : Data from Local System**	15		S	NP-DIS	[9] Assuming espionage type activity

17	**TA0009 : Collection**	**T1074.001 : Data Staged: Local Data Staging**	16		S	AO-COL	[9] From above
18	**TA0011 : Command and Control**	**T1573.001 : Encrypted Channel: Symmetric Cryptography**	17		S	IF-C2C	[9] Example exfiltration communication (and next command)
19	**TA0007 : Discovery**	**T1046 : Network Service Discovery**	18		S	NP-DIS	From ATT&CK group description
20	**TA0007 : Discovery**	**T1135 : Network Share Discovery**	19		S	NP-DIS	From ATT&CK group description
21	**TA0011 : Command and Control**	**T1573.001 : Encrypted Channel: Symmetric Cryptography**	20		S	IF-C2C	[9] Example discovery info exfiltration communication (and next command)
22	**TA0011 : Command and Control**	**T1105 : Ingress Tool Transer**	21		S	IF-C2C	From ATT&CK group description
23	**TA0011 : Command and Control**	**T1573.001 : Encrypted Channel: Symmetric Cryptography**	22		S	IF-C2C	[9] Example tool response communication (and next command)

[1] Tropic Trooper Targets Taiwanese Government and Fossil Fuel Provider With Poison Ivy (paloaltonetworks.com) [2016]

[2] How Operation Tropic Trooper Infiltrates Secret Keepers - Wiadomości bezpieczeństwa (trendmicro.com) [2015] (not from ATT&CK)

[3] Operation Tropic Trooper: Relying on Tried-and-Tested Flaws to Infiltrate Secret Keepers (trendmicro.com) [2015] (from ATT&CK, but link from ATT&CK broken so re found and listed here)

[4] Familiar Feeling: A Malware Campaign Targeting the Tibetan Diaspora Resurfaces - The Citizen Lab [2018]

[5] Anomali Suspects that China-Backed APT Pirate Panda May Be Seeking Access to Vietnam Government Data Center | Anomali [2020]

[6] Tropic Trooper's Back: USBferry Attack Targets Air-gapped Environments (trendmicro.com) [2020]

[7] Deep Analysis of New Poison Ivy Variant (fortinet.com) [2017] (not from ATT&CK)

[8] CVE - CVE-2012-0158 (mitre.org)

[9] PoisonIvy, Software S0012 | MITRE ATT&CK®

Glossary of Terms

This section is work in progress.

Body Of Knowledge ..... Concepts, terms and activities that make up a professional domain, as defined by the relevant learned society or professional association (Body of Knowledge – Wikipedia)

List of References

Bibliography

2022 IEEE Cybermatics Congress. (2022). https://doi.org/10.5121/ijcnc.2015.7101

Abraham, S., & Nair, S. (2015). A Predictive Framework For Cyber Security Analytics Using Attack Graphs. *International Journal of Computer Networks & Communications*, *7*(1), 01–17. https://doi.org/10.5121/ijcnc.2015.7101

Abu, M. S., Selamat, S. R., Ariffin, A., & Yusof, R. (2018). Cyber threat intelligence – Issue and challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, *10*(1), 371–379. https://doi.org/10.11591/ijeecs.v10.i1.pp371-379

Accenture. (2021). *Cyber Threat Intelligence Report 2021*. *2*, 1–26. https://www.accenture.com/_acnmedia/PDF-172/Accenture-2021-Cyber-Threat-Intelligence-Report.pdf#zoom=40

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proc. of 20th International Conference on Very Large Data Bases, {VLDB'94}*, 487–499. https://courses.cs.duke.edu/compsci516/spring16/Papers/AssociationRuleMining.pdf

Ahmed, Y., Asyhari, A. T., & Rahman, M. A. (2021). A Cyber Kill Chain Approach for Detecting Advanced Persistent Threats. *Computers, Materials and Continua*, *67*(2), 2497–2513. https://doi.org/10.32604/cmc.2021.014223

Aksu, M. U., Bicakci, K., Dilek, M. H., Ozbayoglu, A. M., & Tatlı, E. İ. (2018). Automated generation of attack graphs using NVD. *CODASPY 2018 - Proceedings of the 8th ACM Conference on Data and Application Security and Privacy*, *2018-Janua*, 135–142. https://doi.org/10.1145/3176258.3176339

Al-Shaer, E., & Chu, B. (2017). *STIX Analytics-From Threat Information Sharing to Automated Response Secure and Resilient Cyber Ecosystem Industry Workshop Presentation for DHS*. https://secwww.jhuapl.edu/srce-workshop/past-events/2015/docs/presentations/T2 04 UNC - SRCE Presentations.pdf

Al-Shaer, R., Spring, J. M., & Christou, E. (2020). Learning the Associations of MITRE ATT CK Adversarial Techniques. *2020 IEEE Conference on Communications and Network Security, CNS 2020*. https://doi.org/10.1109/CNS48642.2020.9162207

Alavizadeh, H., Jang-Jaccard, J., Enoch, S. Y., Al-Sahaf, H., Welch, I., Camtepe, S. A., & Kim, D. D.

(2022). A Survey on Cyber Situation-Awareness Systems: Framework, Techniques, and Insights. *ACM Computing Surveys*, *55*(5), 1–25. https://doi.org/10.1145/3530809

AlienVault. (2019a). *OSSIM: The Open Source SIEM*. https://www.alienvault.com/products/ossim

AlienVault. (2019b). *The World's First Truly Open Threat Intelligence Community*. https://otx.alienvault.com/

Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2019). A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. *IEEE Communications Surveys & Tutorials*, *PP*(8), 25. https://doi.org/10.1109/COMST.2019.2891891

Althonayan, A., & Andronache, A. (2018). Shifting from information security towards a cybersecurity paradigm. *ACM International Conference Proceeding Series*, 68–79. https://doi.org/10.1145/3285957.3285971

Andrei Brazhuk. (2019). Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries. *International Journal of Open Information Technologies*, *7*(3), 38–41. http://injoit.org/index.php/j1/article/view/686/675

Andress, J., & Winterfield, S. (2011). *Cyber Warfare - Techniques, Tactics and Tools for Security Practitioners* (Second). Elsevier. https://www.elsevier.com/books/cyber-warfare/andress/978-1-59749-637-7

Ansari, M. S., Bartos, V., & Lee, B. (2020). Shallow and Deep Learning Approaches for Network Intrusion Alert Prediction. *Procedia Computer Science*, *171*(2019), 644–653. https://doi.org/10.1016/j.procs.2020.04.070

Applebaum, A. (2016). *Understanding Cyber Adversaries with ATT&CK – The Post-Exploit Threat Model*. http://www.asq509.org/ht/a/GetDocumentAction/i/124672

Applebaum, A., Johnson, S., Limiero, M., & Smith, M. J. (2018). Playbook oriented cyber response. *Proceedings - 2018 National Cyber Summit Research Track, NCS 2018*, 8–15. https://doi.org/10.1109/NCS.2018.00007

Asgarli, E., & Burger, E. (2016). Semantic ontologies for cyber threat sharing standards. *2016 IEEE Symposium on Technologies for Homeland Security, HST 2016*. https://doi.org/10.1109/THS.2016.7568896

Assante, M., & Lee, R. (2015). The Industrial Control System Cyber Kill Chain. *Sans Institute*,

October, 1–22. https://scadahacker.com/library/Documents/White_Papers/SANS - ICS Cyber Kill Chain.pdf

Bahrami, P. N., Dehghantanha, A., Dargahi, T., Parizi, R. M., Choo, K. K. R., & Javadi, H. H. S. (2019). Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures. *Journal of Information Processing Systems*, *15*(4), 865–889. https://doi.org/10.3745/JIPS.03.0126

Bandla, K., & Westcott, D. (2019). *APTnotes data*. https://github.com/aptnotes/data

Banerjee, A., & Davé, R. N. (2004). Validating clusters using the Hopkins statistic. *IEEE International Conference on Fuzzy Systems*, *1*, 149–153. https://doi.org/10.1109/FUZZY.2004.1375706

Barnum, S. (2008). Common attack pattern enumeration and classification (CAPEC) schema description. *Cigital Inc, Http://Capec. Mitre. Org/Documents/ …*, 1–20. http://capec.mitre.org/documents/documentation/CAPEC_Schema_Description_v1.3.pdf

Barzegar, M., & Shajari, M. (2018). Attack scenario reconstruction using intrusion semantics. *Expert Systems with Applications*, *108*, 119–133. https://doi.org/10.1016/j.eswa.2018.04.030

Bedny, G., & Mesiter, D. (1999). Theory of Activity and Situation Awareness. *International Journal of Cognitive Ergonomics*, *3*(1), 63–72. https://doi.org/https://doi.org/10.1207/s15327566ijce0301_5

Bhatt, P., Yano, E. T., & Gustavsson, P. (2014). Towards a framework to detect multi-stage advanced persistent threats attacks. *Proceedings - IEEE 8th International Symposium on Service Oriented System Engineering, SOSE 2014*, 390–395. https://doi.org/10.1109/SOSE.2014.53

Bianco, D. J. (2014). *Enterprise Security Monitoring*. https://www.first.org/resources/papers/conference2014/first_2014_-_bianco-_david_-_enterprise_security_monitoring_20140610.pdf

Bodeau, D. J., Mccollum, C. D., & Fox, D. B. (2018a). *Cyber Threat Modeling : Survey , Assessment , and Representative Framework Authors : Homeland Security Systems Engineering & Development Institute*. *18*.

Bodeau, D. J., Mccollum, C. D., & Fox, D. B. (2018b). *Cyber Threat Modeling: Survey, Assessment, and Representative Framework*. *18*.

https://www.mitre.org/sites/default/files/publications/pr_18-1174-ngci-cyber-threat-modeling.pdf

Bou-Harb, E., Debbabi, M., & Assi, C. (2014). Cyber scanning: A comprehensive survey. *IEEE Communications Surveys and Tutorials*, *16*(3), 1496–1519. https://doi.org/10.1109/SURV.2013.102913.00020

Brazhuk, A. (2019). Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries. *International Journal of Open Information Technologies*, *7*(3), 38–41.

Brazhuk, A. (2021). *Towards automation of threat modeling based on a semantic model of attack patterns and weaknesses*. https://arxiv.org/abs/2112.04231v1

Brazhuk, A. (2022). *Mapping natural language phrases to the attack patterns*. https://www.researchgate.net/profile/Andrei-Brazhuk/publication/362655535_Chapter2_Mapping_natural_language_phrases_to_the_attack_patterns/links/62f65496c6f6732999c635b3/Chapter2-Mapping-natural-language-phrases-to-the-attack-patterns.pdf

Bristol, U. O. (2023). *The Cyber Security Body Of Knowledge*. https://www.cybok.org/

Broadcom. (2023). *Attack Signatures*. https://www.broadcom.com/support/security-center/attacksignatures?#:~:text=An attack signature is a,it displays a Security Alert.

Bromander, S., Jøsang, A., & Eian, M. (2016). Semantic cyberthreat modelling. *CEUR Workshop Proceedings*, *1788*, 74–78. http://ceur-ws.org/Vol-1788/STIDS_2016_A03_Bromander_etal.pdf

Bromiley, M. (2016). Threat Intelligence: What It Is, and How to Use It Effectively | SANS Institute. *SANS Security Insights*, *September*. https://www.sans.org/cyber-security-intelligence/2016/10/11/threat-intelligence-what-it-is-and-how-to-use-it-effectively

Brown, R. (2019). *Information Security Reading Room The Evolution of Cyber Threat Intelligence (CTI): 2019 SANS CTI Survey*. 18. https://www.sans.org/reading-room/whitepapers/analyst/evolution-cyber-threat-intelligence-cti-2019-cti-survey-38790

Brown, R., & Lee, R. M. (2019). *Information Security Reading Room The Evolution of Cyber Threat Intelligence (CTI): 2019 SANS CTI Survey*. 18. www.sans.org/reading-room/whitepapers/analyst/cyber-threat-intelligence-uses-successes-failures-2017-cti-survey-37677,%0Ahttps://www.sans.org/reading-room/whitepapers/analyst/evolution-cyber-threat-intelligence-cti-2019-cti-survey-38790

Brown, S., Gommers, J., & Serrano, O. (2015). From Cyber Security Information Sharing to Threat Management. *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security - WISCS '15*, 43–49. https://doi.org/10.1145/2808128.2808133

Bryant, B. D., & Saiedian, H. (2017). A novel kill-chain framework for remote security log analysis with SIEM software. *Computers and Security*, *67*, 198–210. https://doi.org/10.1016/j.cose.2017.03.003

Bryant, B. D., & Saiedian, H. (2020). Improving SIEM alert metadata aggregation with a novel kill-chain based classification model. *Computers and Security*, *94*. https://doi.org/10.1016/j.cose.2020.101817

Brynielsson, J., Franke, U., & Varga, S. (2016). *Cyber Situational Awareness Testing*. *2*. https://doi.org/https://doi.org/10.1007/978-3-319-38930-1_12

Bullough, B. L., Yanchenko, A. K., Smith, C. L., & Zipkin, J. R. (2017). Predicting exploitation of disclosed software vulnerabilities using open-source data. *IWSPA 2017 - Proceedings of the 3rd ACM International Workshop on Security and Privacy Analytics, Co-Located with CODASPY 2017*, 45–53. https://doi.org/10.1145/3041008.3041009

Burns, J., & MITRE. (2018). *ATT&CK™ content available in STIX™ 2.0 via public TAXII™ 2.0 server*. https://medium.com/mitre-attack/att-ck-content-available-in-stix-2-0-via-public-taxii-2-0-server-317e5c41e214

Cai, Y., Gu, Z., Wang, L., Li, S., & Han, W. (2020). An APT Group Knowledge Model based on MDATA. *ACM International Conference Proceeding Series*, 374–378. https://doi.org/10.1145/3444370.3444600

Caltagirone, S., Pendergast, A., & Betz, C. (2013). The Diamond Model of Intrusion Analysis. *Threat Connect*, *298*(0704), 1–61. https://apps.dtic.mil/sti/pdfs/ADA586960.pdf

Casey, T. (2018). *Threat Agent Library Helps Identify Information Security Risks*. *March*. https://doi.org/10.13140/RG.2.2.30094.46406

Chadza, T., Kyriakopoulos, K. G., & Lambotharan, S. (2020). Analysis of hidden Markov model learning algorithms for the detection and prediction of multi-stage network attacks. *Future Generation Computer Systems*, *108*, 636–649. https://doi.org/10.1016/j.future.2020.03.014

Chan, A. (2007). *An Analysis of Pairwise Sequence Alignment Algorithm Complexities: Needleman-Wunsch, Smith-Waterman, FASTA, BLAST and Gapped BLAST*. http://angel.cs.msu.su/~salnikov/speccourse/bio/articles/Chan.pdf

Chapman, I. M., Leblanc, S. P., & Partington, A. (2011). Taxonomy of cyber attacks and simulation of their effects. *2011 Military Modeling & Simulation Symposium*, 73–80. http://dl.acm.org/citation.cfm?id=2048569

Chen, C., & Qin, Z. (2009). A systolic architecture with linear space complexity for longest common subsequence problem. *ASICON 2009 - Proceedings 2009 8th IEEE International Conference on ASIC*, 33–36. https://doi.org/10.1109/ASICON.2009.5351612

Chen, P., Desmet, L., & Huygens, C. (2014). A Study on Advanced Persistent Threats. *IFIP International Conference on Communications and Multimedia Security*, 63–72. https://doi.org/https://doi.org/10.1007/978-3-662-44885-4_5

Cheng, B. C., Liao, G. T., Huang, C. C., & Yu, M. T. (2011). A novel probabilistic matching algorithm for multi-stage attack forecasts. *IEEE Journal on Selected Areas in Communications*, *29*(7), 1438–1448. https://doi.org/10.1109/JSAC.2011.110809

Cheung, S., Lindqvist, U., & Fong, M. (2003). Modelling Multistep Cyber Attacks for Scenario Recognition. *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2003*, *2*. https://ieeexplore.ieee.org/abstract/document/1194892

Chng, S., Lu, H. Y., Kumar, A., & Yau, D. (2022). Hacker types, motivations and strategies: A comprehensive framework. *Computers in Human Behavior Reports*, *5*, 100167. https://doi.org/10.1016/j.chbr.2022.100167

Choi, S., Choi, J., Yun, J. H., Min, B. G., & Kim, H. C. (2020). Expansion of ICS testbed for security validation based on MITRE ATT&CK techniques. *CSET 2020 - 13th USENIX Workshop on Cyber Security Experimentation and Test, Co-Located with USENIX Security 2020*. https://www.usenix.org/system/files/cset20-paper-choi.pdf

Choi, S., Yun, J. H., & Min, B. G. (2021). Probabilistic Attack Sequence Generation and Execution Based on MITRE ATT&CK for ICS Datasets. *ACM International Conference Proceeding Series*, 41–48. https://doi.org/10.1145/3474718.3474722

Chomiak-orsa, I., Rot, A., & Blaicke, W. (2019). Artificial Intelligence in Cybersecurity: The Use of AI Along the Cyber Kill Chain. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11684 LNAI*. https://doi.org/10.1007/978-3-030-28374-2_29

CIRCL. (2019). *CIRCL MISP Threat Sharing*. https://misppriv.circl.lu/users/login

CIRCL. (2020). *MISP exchange format*. https://github.com/MISP/MISP-rfc

Cisco. (n.d.). *Talos Advisories*. https://www.snort.org/talos

Cloudflare. (2022). *What is a software-defined perimeter (SDP)?* https://www.cloudflare.com/en-gb/learning/access-management/software-defined-perimeter/#:~:text=A software-defined perimeter (SDP) is a way to,on software instead of hardware.

Dandurand, L., & Serrano, O. S. (2013). Towards Improved Cyber Security Information Sharing. *International Conference on Cyber Conflict*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6568369

Danyliw, R. (2016). *The Incident Object Description Exchange Format Version 2* (Vol. 62, Issue 1). IETF. https://doi.org/2070-1721

Dargahi, T., Dehghantanha, A., Bahrami, P. N., Conti, M., Bianchi, G., & Benedetto, L. (2019). A Cyber-Kill-Chain based taxonomy of crypto-ransomware features. *Journal of Computer Virology and Hacking Techniques*, *15*(4), 277–305. https://doi.org/10.1007/s11416-019-00338-7

Dass, S., Datta, P., & Namin, A. S. (2021). Attack prediction using Hidden Markov Model. *Proceedings - 2021 IEEE 45th Annual Computers, Software, and Applications Conference, COMPSAC 2021*, 1695–1702. https://doi.org/10.1109/COMPSAC51774.2021.00253

De Bruijne, M., Van Eeten, M., Hernández Gañán, C., & Pieters, W. (2017). *Towards a new cyber threat actor typology: A hybrid method for the NCSC cyber security assessment*. 1–72. https://www.wodc.nl/binaries/2740_Volledige_Tekst_tcm28-273243.pdf

Debar, H., Curry, D., & Feinstein, B. (2007). *IDMEF*. https://www.ietf.org/rfc/rfc4765.txt

DELL. (2014). *Advanced Threat Protection with Dell SecureWorks Security Services*. https://www.secureworks.com/~/media/Files/US/Solution Briefs/DellSecureWorksNCO346NAdvancedThreatProtection.ashx

Derbyshire, R., Green, B., Prince, D., Mauthe, A., & Hutchison, D. (2018). An Analysis of Cyber Security Attack Taxonomies. *Proceedings - 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018*, 153–161. https://doi.org/10.1109/EuroSPW.2018.00028

Di Bernardino, E., & Brogi, G. (2019). Hidden Markov models for advanced persistent threats. *International Journal of Security and Networks*, *14*(4), 181. https://doi.org/10.1504/ijsn.2019.10024503

Du, H., Murphy, C., Bean, J., & Yang, S. J. (2009). Toward unsupervised classification of non-uniform cyber attack tracks. *2009 12th International Conference on Information Fusion, FUSION 2009*, 1919–1925. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5203866

Dutta, S. (2020). *MALOnt*. https://github.com/aiforsec/MALOnt

E-infrastruktura CESNET. (2020). *Intrusion Detection Extensible Alert*. https://idea.cesnet.cz/en/index

Eckmann, S. T., Vigna, G., & Kemmerer, R. A. (2002). STATL: An attack language for state-based intrusion detection. *Journal of Computer Security*, *10*(1–2), 71–103. https://doi.org/10.3233/JCS-2002-101-204

Ehab Al-shaer, Ehab;Chu, B. (2017). *STIX Analytics-- From Threat Information Sharing to Automated Response*. https://pdfs.semanticscholar.org/presentation/aa40/7f5c2039e042db8763fac64220f94581c9fd.pdf

Elgh, J. (2022). *Comparison of adversary emulation tools for reproducing behavior in cyber attacks*. https://www.diva-portal.org/smash/record.jsf?pid=diva2:1632554

Elitzur, A., Puzis, R., & Zilberman, P. (2019). Attack Hypothesis Generation. *Proceedings of the 2019 European Intelligence and Security Informatics Conference, EISIC 2019*, 40–47. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9108886

Endsley, M. R. (1988). DESIGN AND EVALUATION FOR SITUATION AWARENESS ENHANCEMEN. *Proceedings of the Human Factors Society Annual Meeting*, *32*(2), 96–96. https://doi.org/10.1177/154193128803200220

Endsley, M. R. (1995a). Measurement of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 65–84. https://doi.org/10.1518/001872095779049499

Endsley, M. R. (1995b). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1). https://doi.org/https://doi.org/10.1518/001872095779049543

Engle, G. (2014). *Deconstructing The Cyber Kill Chain*. https://www.darkreading.com/attacks-breaches/deconstructing-the-cyber-kill-chain

ENISA. (2014). Actionable Information for Security Incident Response. In *Enisa* (Issue November). https://www.enisa.europa.eu/publications/actionable-information-for-security/

ENISA. (2021). ENISA Threat Landscape 2021. In *European Union Agency for Cybersecurity* (Issue October). https://doi.org/10.2824/324797

European Union. (2015). Cyber Security Information Sharing : An Overview of Regulatory and Non-regulatory Approaches. In *Online* (Issue December). https://doi.org/10.2824/43639

FireEye. (2019). *FireEye - Threat Research*. https://www.fireeye.com/blog/threat-research.html

Five, C. (2011). *Advanced Persistent Threats : A Decade in Review*. *June*, 1–13. https://paper.seebug.org/papers/APT/APT_CyberCriminal_Campagin/2011/C5_APT_ADecadeInReview.pdf

Franke, U., & Brynielsson, J. (2014). Cyber situational awareness - A systematic review of the literature. *Computers and Security*, *46*, 18–31. https://doi.org/10.1016/j.cose.2014.06.008

Franklin, L., Pirrung, M., Blaha, L., Dowling, M., & Feng, M. (2017). Toward a visualization-supported workflow for cyber alert management using threat models and human-centered design. *2017 IEEE Symposium on Visualization for Cyber Security, VizSec 2017*, *2017-Octob*, 1–8. https://doi.org/10.1109/VIZSEC.2017.8062200

Garvey, M. D. (2021). A Philosophical Examination on the Definition of Cyberspace. *Cyber Security and Supply Chain Management*, 1–11. https://doi.org/10.1142/9789811233128_0001

Gawron, V. J. (2019). Measures of Situational Awareness. In *Human Performance and Situation Awareness Measures* (Issue 15817). https://doi.org/10.1201/9780429001024-3

Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K., & Aparicio-Navarro, F. J. (2018). Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, *89*, 349–359. https://doi.org/10.1016/j.future.2018.06.055

Ghafir, I., & Prenosil, V. (2016). Proposed approach for targeted attacks detection. *Lecture Notes in Electrical Engineering*, *362*, 73–80. https://doi.org/10.1007/978-3-319-24584-3_7

Gibb, W., & Kerr, D. (2013). *OpenIOC: Back to the Basics*. https://www.mandiant.com/resources/blog/openioc-basics

Giura, P., & Wang, W. (2012). A context-based detection framework for advanced persistent threats. *Proceedings of the 2012 ASE International Conference on Cyber Security,*

CyberSecurity 2012, *SocialInformatics*, 69–74. https://doi.org/10.1109/CyberSecurity.2012.16

Greene, T. (2016). *Why the 'cyber kill chain' needs an upgrade*. https://www.networkworld.com/article/3104542/why-the-cyber-kill-chain-needs-an-upgradesecurity-pros-need-to-focus-more-on-catching-attackers-aft.html

Grigorescu, O., Nica, A., Dascalu, M., & Rughinis, R. (2022). *CVE2ATT&CK : BERT-Based Mapping of CVEs to MITRE*. 1–22. https://www.mdpi.com/1999-4893/15/9/314

Gschwandtner, M., Demetz, L., Gander, M., & Maier, R. (2018). Integrating Threat Intelligence to Enhance an Organization's Information Security Management. *Proceedings of the 13th International Conference on Availability, Reliability and Security - ARES 2018*, 1–8. https://doi.org/10.1145/3230833.3232797

Gusfield, D. (1997). Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. *Journal of the American Statistical Association*, *94*(447), 989. https://doi.org/10.2307/2670026

Haas, S., & Fischer, M. (2018). GAC: Graph-based alert correlation for the detection of distributed multi-step attacks. *Proceedings of the ACM Symposium on Applied Computing*, 979–988. https://doi.org/10.1145/3167132.3167239

Halbardier, A., Waltermire, D., & Johnson, M. (2011). *Specification for the Asset Reporting Format v1.1*. https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7694.pdf

Han, K. J., Choi, B. Y., & Song, S. (2013). High performance cloud auditing and applications. *High Performance Cloud Auditing and Applications*, *9781461432*(December 2015), 1–360. https://doi.org/10.1007/978-1-4614-3296-8

Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers and Security*, *24*(1), 31–43. https://doi.org/10.1016/j.cose.2004.06.011

Hart, S., Margheri, A., Paci, F., & Sassone, V. (2020). Riskio: A Serious Game for Cyber Security Awareness and Education. *Computers and Security*, *95*. https://doi.org/10.1016/j.cose.2020.101827

Hemberg, E., Kelly, J., Shlapentokh-Rothman, M., Reinstadler, B., Xu, K., Rutar, N., & O'Reilly, U.-M. (2020). Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting. In *Proceedings of ACM Conference (Conference'17)* (Vol. 1, Issue 1). Association for Computing Machinery.

http://arxiv.org/abs/2010.00533

Hoffmann, R. (2019). Markov Models of Cyber Kill Chains with Iterations. *2019 International Conference on Military Communications and Information Systems, ICMCIS 2019*, 1–6. https://doi.org/10.1109/ICMCIS.2019.8842810

Homer, J., Zhang, S., Ou, X., Schmidt, D., Du, Y., Rajagopalan, S. R., & Singhal, A. (2013). Aggregating vulnerability metrics in enterprise networks using attack graphs. *Journal of Computer Security*, *21*(4), 561–597. https://doi.org/10.3233/JCS-130475

Hopkins, B., & Skelllam, J. (1954). *A New Method for determining the Type of Distribution of Plant Individuals*. *18*(70), 213–227. https://www.jstor.org/stable/pdf/42907238.pdf

Hosangadi, S. (2012). *Distance Measures for Sequences*. 1–16. http://arxiv.org/abs/1208.5713

Howard, R., & Olson, R. (2020). Implementing Intrusion Kill Chain Strategies. *The Cyber Defense Review*, *5*(3), 59–76. https://www.jstor.org/stable/26954873

Husák, M., Komárková, J., Bou-Harb, E., & Čeleda, P. (2019). Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys and Tutorials*, *21*(1), 640–660. https://doi.org/10.1109/COMST.2018.2871866

Husari, G., Al-Shaer, E., Ahmed, M., Chu, B., & Niu, X. (2017). TTPDrill: Automatic and accurate extraction of threat actions from unstructured text of CTI Sources. *ACM International Conference Proceeding Series*, *Part F1325*, 103–115. https://doi.org/10.1145/3134600.3134646

Husari, G., Al-Shaer, E., Chu, B., & Rahman, R. F. (2019). Learning APT chains from cyber threat intelligence. *ACM International Conference Proceeding Series*, 0–1. https://doi.org/10.1145/3314058.3317728

Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Proceedings of the 6th International Conference on Information Warfare and Security*, *July 2005*, 113–125. https://doi.org/10.1103/PhysRevLett.86.1110

Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., Ferragut, E., & Goodall, J. (2015). Developing an ontology for cyber security knowledge graphs. *ACM International Conference Proceeding Series*, *06-08-Apri*(April). https://doi.org/10.1145/2746266.2746278

IBM. (2019). *IBM® X-Force Exchange*. https://exchange.xforce.ibmcloud.com/

IETF. (2017). *The JavaScript Object Notation (JSON) Data Interchange Format*. https://tools.ietf.org/html/rfc8259

Ioannou, G., Louvieris, P., & Clewley, N. (2019). A Markov Multi-Phase Transferable Belief Model for Cyber Situational Awareness. *IEEE Access*, *7*, 39305–39320. https://doi.org/10.1109/ACCESS.2019.2897923

Jajodia, S., & Albanese, M. (2017). *An Integrated Framework For Cyber Situational Awareness*. *July*, 203–226. https://doi.org/10.1007/978-3-319-61152-5_8

Jajodia, S., Liu, P., Swarup, V., & Wang, C. (2010). *Cyber situational awareness: advances in information security*. https://doi.org/doi:10.1007/978- 1-4419-0140-8_1

Jiang, L., Jayatilaka, A., Nasim, M., Grobler, M., Zahedi, M., & Babar, M. A. (2022). Systematic Literature Review on Cyber Situational Awareness Visualizations. *IEEE Access*, *10*, 57525–57554. https://doi.org/10.1109/ACCESS.2022.3178195

Johnson, Pontus; Lagerstrom, Robert; Ekstedt, M. (2014). *MAL (the Meta Attack Language): A Domain Specific Language for Probabilistic Threat Modeling and Attack Simulation Pontus*. *6*, 18–28. https://dl.acm.org/doi/10.1145/3230833.3232799

Johnson, P., Lagerström, R., & Ekstedt, M. (2018). A meta language for threat modeling and attack simulations. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3230833.3232799

Jordan, B. (2016). *TAXII Graphic Resources*. https://freetaxii.github.io/

Ju, A., Gou, Y., & Li, T. (2020). MCKC : a modified cyber kill chain model for cognitive APTs analysis within Enterprise multimedia network. *Springer*, 29923–29949. https://doi.org/https://doi.org/10.1007/s11042-020-09444-x

Kaiafas, G. (European C. (2017). *Horizon 2020. Threat Intelligence Sharing : State of the Art and Requirements*. *700071*, 1–56. https://protective-h2020.eu/wp-content/uploads/2017/07/PROTECTIVE-D5.1-E-0517-Threat-Intelligence-Sharing.pdf

Kaloroumakis, P. E., & Smith, M. J. (2021). *Toward a Knowledge Graph of Cybersecurity Countermeasures*. https://d3fend.mitre.org/resources/D3FEND.pdf

Kampanakis, P. (2014). Security automation and threat information-sharing options. *IEEE Security and Privacy*, *12*(5), 42–51. https://doi.org/10.1109/MSP.2014.99

Kanakogi, K., Washizaki, H., Fukazawa, Y., Ogata, S., Okubo, T., Kato, T., Kanuka, H., Hazeyama, A.,

& Yoshioka, N. (2022). Comparative Evaluation of NLP-Based Approaches for Linking CAPEC Attack Patterns from CVE Vulnerability Information. *Applied Sciences (Switzerland)*, *12*(7). https://doi.org/10.3390/app12073400

Kang, J., Sun, Y., Xie, H., Zhu, X., & Ding, Z. (2021). Analysis System for Security Situation in Cyberspace Based on Knowledge Graph. *Proceedings - 2021 7th International Conference on Big Data and Information Analytics, BigDIA 2021*, *BigDIA*, 385–392. https://doi.org/10.1109/BigDIA53151.2021.9619719

Karafili, E., Wang, L., Kakas, A. C., & Lupu, E. (2018). *Helping Forensic Analysts to Attribute Cyber-Attacks: An Argumentation-Based Reasoner* (Vol. 11224). Springer International Publishing. https://doi.org/10.1007/978-3-030-03098-8

Khairkar, A. D., Kshirsagar, D. D., & Kumar, S. (2013). Ontology for detection of web attacks. *Proceedings - 2013 International Conference on Communication Systems and Network Technologies, CSNT 2013*, 612–615. https://doi.org/10.1109/CSNT.2013.131

Khaleefa, E. J., & Abdulah, D. A. (2022). Concept and difficulties of advanced persistent threats (APT): Survey. *Int. J. Nonlinear Anal. Appl*, *13*(November 2021), 2008–6822. http://dx.doi.org/10.22075/ijnaa.2022.6230

Khan, M. S., Siddiqui, S., Ferens, K., & Chain, C. K. (2018). *A Cognitive and Concurrent Cyber Kill Chain Model*. https://doi.org/10.1007/978-3-319-58424-9

Kigerl, A. (2016). *Cyber Crime Nation Typologies : K-Means Clustering of Countries Based on Cyber Crime Rates*. *10*(2), 147–169. https://doi.org/10.5281/zenodo.163399/

Kim, H., Kwon, H. J., & Kim, K. K. (2019). Modified cyber kill chain model for multimedia service environments. *Multimedia Tools and Applications*, *78*(3), 3153–3170. https://doi.org/10.1007/s11042-018-5897-5

Kim, S., Heo, G., Zio, E., Shin, J., & Song, J. (2019). Cyber attack taxonomy for digital environment in nuclear power plants. *Nuclear Engineering and Technology*, *xxxx*, 1–7. https://doi.org/https://doi.org/10.1016/j.net.2019.11.001

Kirillov, I. A., Beck, D., Chase, M. P., & Martin, R. (2015). *The Concepts of the Malware Attribute Enumeration and Characterization (MAEC) Effort*. 1–20. https://maec.mitre.org/about/docs/The_MAEC_Concept.pdf

Koepke, P. (2017). *Cybersecurity Information Sharing Incentives and Barriers*. *June*. http://web.mit.edu/smadnick/www/wp/2017-13.pdf

Koilpillai, juanita. (2019). *Will the Software Defined Perimeter Debunk the Cyber Kill Chain?* https://www.waverleylabs.com/will-the-software-defined-perimeter-debunk-the-cyber-kill-chain/

Komárková, J., Husák, M., & Tovarňák, D. (2018). CRUSOE : Data Model for Cyber Situational Awareness. *Proceedings of the 13th International Conference on Availability, Reliability and Security - ARES 2018*. https://doi.org/10.1145/3230833.3232798

Kuppa, A., Aouad, L., & Le-Khac, N.-A. (2021). Linking CVE's to MITRE ATT&CK Techniques. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3465481.3465758

Kurniawan, K., Ekelhart, A., & Kiesling, E. (2021). An att&ck-kg for linking cybersecurity attacks to adversary tactics and techniques? *CEUR Workshop Proceedings*, *2980*, 3–7. http://eprints.cs.univie.ac.at/7202/1/ISWC2021_Poster_Attack_Andreas Ekelhart.pdf

Kurniawan, K., Ekelhart, A., Kiesling, E., Quirchmayr, G., & Tjoa, A. M. (2022). KRYSTAL : Knowledge graph-based framework for tactical attack discovery in audit data. *Computers & Security*, *121*, 102828. https://doi.org/10.1016/j.cose.2022.102828

Kurniawan, K., Ekelhart, A., Kiesling, E., Winkler, D., Quirchmayr, G., & Tjoa, A. M. (2022). VloGraph: A Virtual Knowledge Graph Framework for Distributed Security Log Analysis. *Machine Learning and Knowledge Extraction*, *4*(2), 371–396. https://doi.org/10.3390/make4020016

Kwok, R. (2019). *Viterbi algorithm for prediction with HMM — Part 3 of the HMM series*. https://medium.com/analytics-vidhya/viterbi-algorithm-for-prediction-with-hmm-part-3-of-the-hmm-series-6466ce2f5dc6

Ladisa, P., Plate, H., Martinez, M., & Barais, O. (2022). *Taxonomy of Attacks on Open-Source Software Supply Chains*. 1–18. http://arxiv.org/abs/2204.04008

Laliberte, M. (2016). *A Twist On The Cyber Kill Chain: Defending Against A JavaScript Malware Attack*. https://www.darkreading.com/attacks-breaches/a-twist-on-the-cyber-kill-chain-defending-against-a-javascript-malware-attack

Lallie, H. S., Debattista, K., & Bal, J. (2020). A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*, *35*, 100219. https://doi.org/10.1016/j.cosrev.2019.100219

Lange, M., Kott, A., Ben-Asher, N., Mees, W., Baykal, N., Vidu, C.-M., Merialdo, M., Malowidzki, M., & Madahar, B. (2017). Recommendations for Model-Driven Paradigms for Integrated

Approaches to Cyber Defense. *Article in International Journal of Electrical Engineering Education*, *abs/1703.0*. http://arxiv.org/abs/1703.03306

Lawson, R. G., & C, J. P. (1990). New Index for Clustering Tendency and Its Application to Chemical Problems. *J. Chem. Inf. Comput. Sci. 1990, 30, 1, 36–41*, 941–949. https://doi.org/https://doi.org/10.1021/ci00065a010

Legoy, V., Caselli, M., Seifert, C., & Peter, A. (2020). *Automated Retrieval of ATT&CK Tactics and Techniques for Cyber Threat Reports*. 1–20. https://arxiv.org/pdf/2004.14322.pdf

Lemay, A., Calvet, J., Menet, F., & Fernandez, J. M. (2018). Survey of publicly available reports on advanced persistent threat actors. *Computers and Security*, *72*, 26–59. https://doi.org/10.1016/j.cose.2017.08.005

Li, M., Huang, W., Wang, Y., Fan, W., & Li, J. (2016). The study of APT attack stage model. *2016 IEEE/ACIS 15th International Conference on Computer and Information Science, ICIS 2016 - Proceedings*, 0–4. https://doi.org/10.1109/ICIS.2016.7550947

Li, T., Liu, Y., Liu, Y., Xiao, Y., & Nguyen, N. A. (2020). Attack plan recognition using hidden Markov and probabilistic inference. *Computers and Security*, *97*, 101974. https://doi.org/10.1016/j.cose.2020.101974

Lin, S.-X., Li, Z.-J., Chen, T.-Y., & Wu, D.-J. (2021). Attack Tactic Labeling for Cyber Threat Hunting. *International Conference on Advanced Communications Technology(ICACT)*, *1*, 34–39. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9728949

Linddun. (2022). *LINDDUN privacy engineering*. https://www.linddun.org/

Lindqvist, U., & Porras, P. A. (1999). Detecting computer and network misuse through the production-based expert system toolset (P-BEST). *Proceedings - IEEE Symposium on Security and Privacy*, *1999-Janua*, 146–161. https://doi.org/10.1109/SECPRI.1999.766911

Liu, P., Jajodia, S., & Wang, C. (2017). *Theory and Models for Cyber Situation Awareness*. https://www.springer.com/gp/book/9783319611518

Lockheed Martin Corporation. (2015). *Gaining the Advantage - Applying Cyber Kill Chain Methodology to Network Defense*. 1–13. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/Gaining_the_Advantage_Cyber_Kill_Chain.pdf

Luiijf, E., Klaver, M., Luiijf, E., Klaver, M., Security, C., Mason, I., & Shenoi, S. (2017). *On the Sharing*

of Cyber Security Information. https://link.springer.com/chapter/10.1007/978-3-319-26567-4_3

Lusthaus, J., Bruce, M., & Phair, N. (2020). Mapping the Geography of Cybercrime: A Review of Indices of Digital Offending by Country. *Proceedings - 5th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2020*, 448–453. https://doi.org/10.1109/EuroSPW51379.2020.00066

Maidens, C. (2023). *Thesis Dataset Pure ID 160918257*. https://doi.org/https://doi.org/10.5258/SOTON/D2912

Malone, S. T. (2016). Using an Expanded Cyber Kill Chain Model To Increase Attack Resiliency. *Black Hat USA 2016*. https://www.blackhat.com/docs/us-16/materials/us-16-Malone-Using-An-Expanded-Cyber-Kill-Chain-Model-To-Increase-Attack-Resiliency.pdf

Mandiant. (n.d.). *China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media OutletsNo Title*. https://www.mandiant.com/resources/china-based-threat

Mandiant. (2013). *Mandiant APT 1 report*. 76. https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf

Marks, P. (2011). *Dot-dash-diss: The gentleman hacker's 1903 lulz*. New Scientist. https://www.newscientist.com/article/mg21228440-700-dot-dash-diss-the-gentleman-hackers-1903-lulz/

Martin, T. (2021). On the Need for Collaborative Intelligence in Cybersecurity. *CEUR Workshop Proceedings*, *3125*, 100–112. file:///C:/Users/chris/OneDrive - University of Southampton/ResearchProposalShare/electronics-11-02067-v2.pdf

Maschmeyer, L., Deibert, R. J., & Lindsay, J. R. (2020). A tale of two cybers - how threat reporting by cybersecurity firms systematically underrepresents threats to civil society. *Journal of Information Technology and Politics*, *00*(00), 1–20. https://doi.org/10.1080/19331681.2020.1776658

Matheus, C. J., Kokar, M., & Baclawski, K. (2003). *A Core Ontology for Situation Awareness*. *July*, 545–552. https://doi.org/10.1109/ICIF.2003.177494

Mavroeidis, V., & Bromander, S. (2017). Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. *Proceedings - 2017 European Intelligence and Security Informatics Conference, EISIC 2017*, *2017-Janua*, 91–

98. https://doi.org/10.1109/EISIC.2017.20

Mavroeidis, V., Hohimer, R., Casey, T., & Jesang, A. (2021). Threat Actor Type Inference and Characterization within Cyber Threat Intelligence. *International Conference on Cyber Conflict, CYCON*, *2021-May*(303585), 327–352. https://doi.org/10.23919/CyCon51939.2021.9468305

Maymí, F., Bixler, R., Jones, R., & Lathrop, S. (2018). Towards a definition of cyberspace tactics, techniques and procedures. *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017*, *2018-Janua*, 4674–4679. https://doi.org/10.1109/BigData.2017.8258514

Maynard, P., McLaughlin, K., & Sezer, S. (2020). Decomposition and sequential-and analysis of known cyber-attacks on critical infrastructure control systems. *Journal of Cybersecurity*, *6*(1), 1–20. https://doi.org/10.1093/CYBSEC/TYAA020

Mcguinness, B. (1999). Situational awareness and the CREW awareness rating scale (CARS). *1999 Avionics Conference*.

Mead, N. R., Shull, F., Vemuru, K., & Villadsen, O. (2018). A Hybrid Threat Modeling Method. *Carnegie Mellon University Software Engineering Institute*, *March*, 41. https://resources.sei.cmu.edu/asset_files/TechnicalNote/2018_004_001_516627.pdf

Meyers, C., Powers, S., & Faissol, D. (2009). Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches. *… National Laboratory (April …*, *August 2007*, 1–22. https://doi.org/10.2172/967712

Microsoft. (2022). *The STRIDE Threat Model*. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN

Miller, D., Alford, R., Applebaum, A., Foster, H., Little, C., & Strom, B. (2018). *Automated adversary emulation: A case for planning and acting with unknowns*. 9. https://www.mitre.org/sites/default/files/publications/pr-18-0944-1-automated-adversary-emulation-planning-acting.pdf

Mireles, J. D., Cho, J., & Xu, S. (2016). *Extracting Attack Narratives from Traffic Datasets*. 0–5. https://doi.org/10.1109/CYCONUS.2016.7836624

MISP. (2018a). *MISP - Open Source Threat Intelligence Platform & Open Standards For Threat Information Sharing*. http://www.misp-project.org/

MISP. (2018b). *MISP - User guide*. https://circl.lu/doc/misp/book.pdf

MISP. (2019). Best Practices in Threat Intelligence. *GitHub*, 15. https://github.com/MISP/best-practices-in-threat-intelligence

MIT. (2011). *Matching DNA Sequences*. 4–7. https://courses.csail.mit.edu/6.006/fall11/rec/rec09b.pdf

MITRE. (n.d.-a). *Adversary Emulation Library*. https://github.com/center-for-threat-informed-defense/adversary_emulation_library

MITRE. (n.d.-b). *CWE Common Weakness Enumeration*. http://cwe.mitre.org/

MITRE. (2019a). *CAPEC - Common Attack Pattern Enumeration & Classification*. https://capec.mitre.org/

MITRE. (2019b). *Media Resources*. https://www.mitre.org/news/media-resources

MITRE. (2019c). *MITRE ATT&CK*. https://attack.mitre.org/

MITRE. (2019d). *Threat Report ATT&CK Mapper (TRAM)*. https://github.com/mitre-attack/tram

MITRE. (2022a). *Accessing MITRE ATT&CK data*. https://github.com/mitre/cti/blob/master/USAGE.md

MITRE. (2022b). *ATT&CK Data Sources*. https://github.com/mitre-attack/attack-datasources

MITRE. (2022c). *Cyber Threat Intelligence Repository expressed in STIX 2.0*. https://github.com/mitre/cti

MITRE. (2022d). *ICS Matrix*. https://attack.mitre.org/matrices/ics/

MITRE. (2022e). *Mapping ATT&CK to CVE for Impact*. https://ctid.mitre-engenuity.org/our-work/attck-to-cve/

MITRE. (2022f). *MITRE ATT&CK TAXII Service*. https://cti-taxii.mitre.org/taxii/

MITRE. (2022g). *MITRE Cyber Analytics Repository*. https://car.mitre.org/

MITRE. (2022h). *MITRE D3FEND*. https://d3fend.mitre.org/

Myagmar, S., Lee, A. J., & Yurcik, W. (2005). Threat Modeling as a Basis for Security Requirements. *Proceedings of the 2005 ACM Workshop on Storage Security and Survivability*, 94–102. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.8462&rep=rep1&type=pdf

Nachreiner, C. (2015). *Kill Chain 3.0: Update the cyber kill chain for better defense*.

https://www.helpnetsecurity.com/2015/02/10/kill-chain-30-update-the-cyber-kill-chain-for-better-defense/

Naik, N., Jenkins, P., Grace, P., & Song, J. (2022). Comparing Attack Models for IT Systems : Lockheed Martin's Cyber Kill Chain , MITRE ATT&CK Framework and Diamond Model. *2022 IEEE International Symposium on Systems Engineering (ISSE)*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10005490

Narayanan, S. N., Ganesan, A., Joshi, K., Oates, T., Joshi, A., & Finin, T. (2018). Early detection of cybersecurity threats using collaborative cognition. *Proceedings - 4th IEEE International Conference on Collaboration and Internet Computing, CIC 2018*, 354–363. https://doi.org/10.1109/CIC.2018.00054

National Cyber Security Centre. (2016). *Cyber Security Assessment Netherlands csan 2016*. https://securitydelta.nl/media/com_hsd/report/105/document/CSAN2016.pdf

National Cyber Security Centre. (2021). *Cyber Security Assessment Netherlands 2021*. 1–84. https://english.nctv.nl/documents/publications/2021/08/05/cyber-security-assessment-netherlands-2021

NCSC. (2015). *Understanding vulnerabilities*. https://www.ncsc.gov.uk/information/understanding-vulnerabilities

NCSC. (2019). *Threat reports*. https://www.ncsc.gov.uk/section/keep-up-to-date/threat-reports

NERC. North American Electric Reliability Corporation. (2010). *Security Guideline for the Electricity Sector: Identifying Critical Cyber Assets*. 1–47. http://www.nerc.com/docs/cip/sgwg/Critcal_Cyber_Asset_ID_V1_Final.pdf

Neto, A. J. H., Dos Santos, A. F. P., & Dos Santos, M. (2021). Polymer: An Adaptive Kill Chain Expanding Cyber Threat Hunting to Multi-Platform Environments. *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, 2128–2135. https://doi.org/10.1109/BigData52589.2021.9671731

Niakanlahiji, A., Wei, J., & Chu, B. T. (2019). A Natural Language Processing Based Trend Analysis of Advanced Persistent Threat Techniques. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 2995–3000. https://doi.org/10.1109/BigData.2018.8622255

Nicholas, P. (2017). *Open information sharing is key to tackling cyber security*. https://www.microsoft.com/en-us/cybersecurity/blog-hub/information-sharing-key-to-cybersecurity

Ning, H., Li, Q., Wei, D., Liu, H., & Zhu, T. (2017). Cyberlogic Paves the Way from Cyber Philosophy to Cyber Science. *IEEE Internet of Things Journal*, *4*(3), 783–790. https://doi.org/10.1109/JIOT.2017.2666798

Ning, H., Ye, X., Bouras, M. A., Wei, D., & Daneshmand, M. (2018). General cyberspace: Cyberspace and cyber-enabled spaces. *IEEE Internet of Things Journal*, *5*(3), 1843–1856. https://doi.org/10.1109/JIOT.2018.2815535

NIST. (2013). Glossary of key information security terms. *Nist Ir*, *7298*(Revision 2), 222. http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

NIST. (2019a). Glossary of key information security terms. *Nist Ir*, *Revision 3*, 11. https://doi.org/10.6028/NIST.IR.7298r3

NIST. (2019b). *JSON Schema for NVD Vulnerability Data Feed version 1.0*. https://csrc.nist.gov/schema/nvd/feed/1.0/nvd_cve_feed_json_1.0.schema

NIST. (2019c). *NVD - Home*. https://nvd.nist.gov/

NIST. (2019d). *Security Content Automation Protocol (SCAP)*. https://csrc.nist.gov/projects/security-content-automation-protocol

NIST. (2023). *APT*. https://csrc.nist.gov/glossary/term/apt

Noel, S., & Heinbockel, W. (2015). An Overview of MITRE Cyber Situational Awareness Solutions. *Mitre Corporation*, *May*, 1–17.

Noor, U., Anwar, Z., Amjad, T., & Choo, K. K. R. (2019). A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise. *Future Generation Computer Systems*, *96*, 227–242. https://doi.org/10.1016/j.future.2019.02.013

Noor, U., Anwar, Z., & Rashid, Z. (2018). An Association Rule Mining-Based Framework for Profiling Regularities in Tactics Techniques and Procedures of Cyber Threat Actors. *2018 International Conference on Smart Computing and Electronic Enterprise, ICSCEE 2018*, 1–6. https://doi.org/10.1109/ICSCEE.2018.8538379

OASIS. (2017). *STIX Version 2.0. Part 1: STIX Core Concepts*. *July*, 1–71. http://docs.oasis-open.org/cti/stix/v2.0/cs01/part1-stix-core/stix-v2.0-cs01-part1-stix-core.pdf

OASIS. (2018). *Sharing threat intelligence just got a lot easier*. https://oasis-open.github.io/cti-documentation/

Obrst, L., Chase, P., & Markeloff, R. (2014). Developing an ontology of the cyber security domain. *CEUR Workshop Proceedings*, *966*, 49–56. https://doi.org/10.2307/190645

OffensiveSecurity. (2019). *Exploit Database*. https://www.exploit-db.com/

Onwubiko, C. (2017). Understanding Cyber Situation Awareness. *International Journal on Cyber Situational Awareness*, *1*(1), 11–30. https://doi.org/10.22619/ijcsa.2016.100101

Otten, N. van. (2023). *What is SimHash?* https://spotintelligence.com/2023/01/02/simhash/

Ottis, R., & Lorents, P. (1984). *Cyberspace : Definition and Implications*. https://search.proquest.com/docview/869617247?pq-origsite=gscholar

Panda Security. (2017). Understanding Cyber-attacks. Part I | 2. *Intelligence Platform*. http://resources.pandasecurity.com/enterprise/solutions/ad360/1704-WHITEPAPER-CKC-EN.pdf

Parmelee, M. C. (2010). Toward an ontology architecture for cyber-security standards. *CEUR Workshop Proceedings*, *713*. https://stids.c4i.gmu.edu/STIDS2010/presentations/STIDS_talk_A8_Parmelee.pdf

Pedrinaci, C., & Domingue, J. (2010). Toward the Next Wave of Services: Linked Services for the Web of Data. *Journal Of Universal Computer Science*, *16*(13), 1694–1719. https://doi.org/10.1145/2020408.2020428

Pöhn, D., & Hommel, W. (2022). *TaxIdMA: Towards a Taxonomy for Attacks related to Identities*. 1–13. https://doi.org/10.1145/3538969.3544430

Pokorno, Z., Barysevich, A., Gundert, L., Liska, A., McDaniel, M., Wetzel, J., & Ahlberg, C. (2019). *The Threat Intelligence Handbook*. https://paper.bobylive.com/Security/threat-intelligence-handbook-second-edition.pdf

Pols, P. (2017). *The Unified Kill Chain*. 1–104. https://www.csacademy.nl/images/scripties/2018/Paul_Pols_-_The_Unified_Kill_Chain_1.pdf

Potteiger, B., Martins, G., & Koutsoukos, X. (2016). Software and Attack Centric Integrated Threat Modeling for Quantitative Risk Assessment. *HotSos '16: Proceedings of the Symposium and Bootcamp on the Science of Security*, 99–108. https://slate.com/arts/2018/01/seth-meyers-looks-into-trumps-dhttps://dl.acm.org/doi/pdf/10.1145/2898375.2898390

Quintero-Bonilla, S., & del Rey, A. M. (2020). A new proposal on the advanced persistent threat: A

survey. *Applied Sciences (Switzerland)*, *10*(11). https://doi.org/10.3390/app10113874

Rahman, M. R., Mahdavi-Hezaveh, R., & Williams, L. (2020). A Literature Review on Mining Cyberthreat Intelligence from Unstructured Texts. *IEEE International Conference on Data Mining Workshops, ICDMW*, *2020-Novem*, 516–525. https://doi.org/10.1109/ICDMW51313.2020.00075

Ramirez, R., & Choucri, N. (2016). Improving Interdisciplinary Communication with Standardized Cyber Security Terminology: A Literature Review. *IEEE Access*, *4*, 2216–2243. https://doi.org/10.1109/ACCESS.2016.2544381

Ramsdale, A., Shiaeles, S., & Kolokotronis, N. (2020). A comparative analysis of cyber-threat intelligence sources, formats and languages. *Electronics (Switzerland)*, *9*(5). https://doi.org/10.3390/electronics9050824

Rastogi, N., Dutta, S., Zaki, M. J., Gittens, A., & Aggarwal, C. (2020). MALOnt: An Ontology for Malware Threat Intelligence. *Communications in Computer and Information Science*, *1271 CCIS*, 28–44. https://doi.org/10.1007/978-3-030-59621-7_2

Raulerson, E. L. (2013). *Modeling Cyber Situational Awareness through Data Fusion*. https://scholar.afit.edu/etd/898/

Rdocumentation.org. (2023). *NbClust: NbClust Package for determining the best number of clusters*. https://www.rdocumentation.org/packages/NbClust/versions/3.0.1/topics/NbClust

Rege, A., Singer, B., Masceri, N., & Heath, Q. (2017). Measuring cyber intrusion chains, adaptive adversarial behavior, and group dynamics. *Proceedings of the 12th International Conference on Cyber Warfare and Security, ICCWS 2017*, 285–294. https://www.proquest.com/docview/1897683939/fulltextPDF/8A3B157679434FDFPQ/5?accountid=13963

Rid, T., & Buchanan, B. (2015). Attributing Cyber Attacks. *Journal of Strategic Studies*, *38*(1–2), 4–37. https://doi.org/10.1080/01402390.2014.977382

Rizov, V. (2018). Information Sharing For Cyber Threats. *Information & Security: An International Journal*, 43–50. https://doi.org/https://doi.org/10.11610/isij.3904

Rodriguez, R. (2022). *ATT&CK Python Client*. https://github.com/OTRF/ATTACK-Python-Client

Roesch, M. (2019). *SNORT Users Manual 2.9.13 The Snort Project*. https://snort-org-site.s3.amazonaws.com/production/document_files/files/000/000/177/original/snort_manu

al.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIXACIED2SPMSC7GA%2F20190514%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190514T203843Z&X-A

Rot, A., & Olszewski, B. (2017). Advanced Persistent Threats Attacks in Cyberspace. Threats, Vulnerabilities, Methods of Protection. *Position Papers of the 2017 Federated Conference on Computer Science and Information Systems*, *12*, 113–117. https://doi.org/10.15439/2017f488

Saini, V., Duan, Q., & Paruchuri, V. (2008). Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, *23*(4), 124–131. http://dl.acm.org/citation.cfm?id=1352100

Satvat, K., Gjomemo, R., & Venkatakrishnan, V. N. (2021). Extractor: Extracting attack behavior from threat reports. *Proceedings - 2021 IEEE European Symposium on Security and Privacy, Euro S and P 2021*, 598–615. https://doi.org/10.1109/EuroSP51992.2021.00046

Sauerwein, C., Sillaber, C., Mussmann, A., & Breu, R. (2017). Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. *Proceedings of the 13th International Conference on Wirtschaftsinformatik (WI 2017)*, 837–851. https://www.wi2017.ch/images/wi2017-0188.pdf

Schlette, D., Böhm, F., Caselli, M., & Pernul, G. (2020). Measuring and visualizing cyber threat intelligence quality. *International Journal of Information Security*. https://doi.org/10.1007/s10207-020-00490-y

Schlette, D., Caselli, M., & Pernul, G. (2021). A Comparative Study on Cyber Threat Intelligence: The Security Incident Response Perspective. *IEEE Communications Surveys and Tutorials*, *23*(4), 2525–2556. https://doi.org/10.1109/COMST.2021.3117338

Schneier, B. (1999). Attack Trees. *Dr Dobbs*. http://macs.citadel.edu/baniks/427/Homework/attacktrees.pdf

Sexton, J., Storlie, C., & Neil, J. (2015). Attack chain detection. *Statistical Analysis and Data Mining*, *8*(5–6), 353–363. https://doi.org/10.1002/sam.11296

Shallabi, N. (2019). *ATT&CK - Tools*. https://github.com/nshalabi/ATTACK-Tools

Shalyapin, A., & Zhukov, V. (2015). *Case based analysis in information security incidents management system*. https://dl.acm.org/doi/pdf/10.1145/2799979.2799990

Shawly, T., Elghariani, A., Kobes, J., & Ghafoor, A. (2018). Architectures for Detecting Real-time

Multiple Multi-stage Network Attacks Using Hidden Markov Model. *Cornell University Library ArXiv*, 1–29. https://arxiv.org/pdf/1807.09764.pdf

Shawly, T., Elghariani, A., Kobes, J., & Ghafoor, A. (2021). *Architectures for Detecting Interleaved Multi-Stage Network Attacks Using Hidden Markov Models*. *18*(5), 2316–2330. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8880499

Shevchenko, N., Chick, T. A., Riordan, P. O., Scanlon, T. P., & Woody, C. (2018). Threat Modeling : a Summary of Available Methods. *Research Report*, *July*, 26. https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf

Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.

Siddiqi, A. ., & Ghani, N. (2016). Critical Analysis on Advanced Persistent Threats. *International Journal of Computer Applications*, *141*(13), 46–50. https://doi.org/10.5120/ijca2016909784

Sigma. (2022). *Generic Signature Format for SIEM Systems*. https://github.com/SigmaHQ/sigma

Sillaber, C., Sauerwein, C., Mussmann, A., & Breu, R. (2016). Data Quality Challenges and Future Research Directions in Threat Intelligence Sharing Practice. *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security - WISCS'16*, 65–70. https://doi.org/10.1145/2994539.2994546

Simmons, C., Shiva, S., Bedi, H., & Dasgupta, D. (2009). AVOIDIT: A Cyber Attack Taxonomy. *9th Annual Symposium on Information Assurance*, 39–47. http://www.teraits.com/pitagoras/marcio/segapp/CyberAttackTaxonomy_IEEE_Mag.pdf

Singh, S., & Silakari, S. (2009). A survey of cyber attack detection systems. *International Journal of Security and Its Applications*, *8*(1), 247–256. https://doi.org/10.14257/ijsia.2014.8.1.23

Skopik, F., Settanni, G., & Fiedler, R. (2016). A problem shared is a problem halved: A survey on the dimensions of collective cyber defense through security information sharing. *Computers and Security*, *60*, 154–176. https://doi.org/10.1016/j.cose.2016.04.003

Smith, K., & Hancock, P. A. (1995). Situation Awareness Is Adaptive, Externally Directed Consciousness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 137–148. https://doi.org/https://doi.org/10.1518/001872095779049444

Soleymanzadeh, R., & Kashef, R. (2022). The Future Roadmap for Cyber-attack Detection. *Proceedings - 2022 6th International Conference on Cryptography, Security and Privacy, CSP 2022*, 66–70. https://doi.org/10.1109/CSP55486.2022.00021

Spring, J. M., & Al-shaer, R. (2020). *Automating reasoning with ATT&CK*. 1–16. https://apps.dtic.mil/sti/pdfs/AD1088924.pdf

Stanton, N. A., Chambers, P. R. G., & Piggott, J. (2001). Situational awareness and safety. *Safety Science*, *39*(3), 189–204. https://doi.org/10.1016/S0925-7535(01)00010-8

Steinberg, A. N., Bowman, C. L., & White, F. E. (1999). Revisions to the JDL data fusion model. *Sensor Fusion: Architectures, Algorithms, and Applications III*, *3719*, 430. https://doi.org/10.1117/12.341367

stillion, ryan. (2014). *On TTPs*. http://ryanstillions.blogspot.com/2014/04/on-ttps.html

Straub, J. (2020). Modeling Attack, Defense and Threat Trees and the Cyber Kill Chain, ATT&CK and STRIDE Frameworks as Blackboard Architecture Networks. *Proceedings - 2020 IEEE International Conference on Smart Cloud, SmartCloud 2020*, 148–153. https://doi.org/10.1109/SmartCloud49737.2020.00035

Strom, B. E., Battaglia, J. A., Kemmerer, M. S., Kupersanin, W., Miller, D. P., Wampler, C., Whitley, S. M., & Wolf, R. D. (2017). *Finding Cyber Threats with ATT&CK-Based Analytics*. *June*, 1–47. https://www.mitre.org/sites/default/files/publications/16-3713-finding-cyber-threats with att%26ck-based-analytics.pdf

Strom, B. E., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2020). *MITRE ATT&CK: Design and Philosophy*. *July 2018*. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf

Studer, M., & Ritschard, G. (2016). What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. *Journal of the Royal Statistical Society. Series A: Statistics in Society*, *179*(2), 481–511. https://doi.org/10.1111/rssa.12125

Suryotrisongko, H., & Musashi, Y. (2019). Review of cybersecurity research topics, taxonomy and challenges: Interdisciplinary perspective. *Proceedings - 2019 IEEE 12th Conference on Service-Oriented Computing and Applications, SOCA 2019*, 162–167. https://doi.org/10.1109/SOCA.2019.00031

Syed, Z., Finin, T., Mathews, L., Joshi, A., & Padia, A. (2016). UCO: A Unified Cybersecurity Ontology. *The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence*, *Artificial*(Association for the Advancement of Artificial Intelligence), 195–202. https://doi.org/10.5057/jjske.TJSKE-D-15-00047

Syed, Z., Padia, A., Finin, T., Mathews, L., & Joshi, A. (2018). *Unified-Cybersecurity-Ontology*.

https://github.com/Ebiquity/Unified-Cybersecurity-Ontology

Takahashi, Y., Shima, S., Tanabe, R., & Yoshioka, K. (2020). APTGen: An approach towards generating practical dataset labelled with targeted attack sequences. *CSET 2020 - 13th USENIX Workshop on Cyber Security Experimentation and Test, Co-Located with USENIX Security 2020*. https://www.usenix.org/system/files/cset20-paper-takahashi.pdf

Tatam, M., Shanmugam, B., Azam, S., & Kannoorpatti, K. (2021). A review of threat modelling approaches for APT-style attacks. *Heliyon*, *7*(1), e05969. https://doi.org/10.1016/j.heliyon.2021.e05969

ThaiCert. (2023). *ThaiCert - Threat Group Cards: A Threat Actor Encyclopedia*. https://apt.etda.or.th/cgi-bin/aptgroups.cgi

Toker, F. S., Akpinar, K. O., & Ozcelik, I. (2021). MITRE ICS Attack Simulation and Detection on EtherCAT Based Drinking Water System. *9th International Symposium on Digital Forensics and Security, ISDFS 2021*, *118*. https://doi.org/10.1109/ISDFS52919.2021.9486331

Tounsi, W., & Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers and Security*, *72*, 212–233. https://doi.org/10.1016/j.cose.2017.09.001

Tundis, A., Ruppert, S., & Mühlhäuser, M. (2022). A Feature-driven Method for Automating the Assessment of OSINT Cyber Threat Sources. *Computers and Security*, *113*, 102576. https://doi.org/10.1016/j.cose.2021.102576

Ucedavélez, T., & Morana, M. M. (2015). Risk Centric Threat Modeling. In *Risk Centric Threat Modeling*. https://doi.org/10.1002/9781118988374

Ulicny, B. E., Moskal, J. J., Kokar, M. M., Abe, K., & Smith, J. K. (2014). Inference and ontologies. *Advances in Information Security*, *62*, 167–199. https://doi.org/10.1007/978-3-319-11391-3_9

Uma, M., & Padmavathi, G. (2013). A survey on various cyber attacks and their classification. *International Journal of Network Security*, *15*(5), 390–396. http://ijns.jalaxy.com.tw/contents/ijns-v15-n5/ijns-2013-v15-n5-p390-396.pdf

University, C. M. (2019). *Vulnerability Notes Database*. https://kb.cert.org/vuls/

Ussath, M., Jaeger, D., Cheng, F., & Meinel, C. (2016). *Advanced Persistent Threats : Behind the Scenes*. 0–5. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7460498

Van Heerden, R., Von Soms, S., & Mooi, R. (2016). Classification of cyber attacks in South Africa. *2016 IST-Africa Conference, IST-Africa 2016*, 1–16. https://doi.org/10.1109/ISTAFRICA.2016.7530663

Varonis. (2018). *What is The Cyber Kill Chain and How to Use it Effectively*. https://www.varonis.com/blog/cyber-kill-chain/

Verizon. (n.d.). *2022 Data Breach Investigations Report*. https://www.verizon.com/business/resources/reports/dbir/2022/results-and-analysis-intro-to-patterns/

Verizon. (2022). *VERIS the vocabulary for event recording and incident sharing*. http://veriscommunity.net/index.html

Villalón-Huerta, A., Ripoll-Ripoll, I., & Marco-Gisbert, H. (2022). A Taxonomy for Threat Actors' Delivery Techniques. *Applied Sciences (Switzerland)*, *12*(8), 1–23. https://doi.org/10.3390/app12083929

Visser, I., & Speekenbrink, M. (2022). *Hidden Markov Models*. 125–172. https://doi.org/10.1007/978-3-031-01440-6_4

Vlachos, M., Kollios, G., & Gunopulos, D. (2002). Discovering similar multidimensional trajectories. *Proceedings - International Conference on Data Engineering*, 673–684. https://doi.org/10.1109/ICDE.2002.994784

Wagner, C., Dulaunoy, A., Wagener, G., & Iklody, A. (2016). MISP - The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security - WISCS'16*, 49–56. https://doi.org/10.1145/2994539.2994542

Waltermire, D., & Fitzgerald-Mckay, J. (2018). *Transitioning to the Security Content Automation Protocol (SCAP) Version 2*. 20. https://doi.org/10.6028/NIST.CSWP.09102018

Wang, W., Tang, B. F., Zhu, C., Liu, B., Li, A., & Ding, Z. (2020). Clustering using a similarity measure approach based on semantic analysis of adversary behaviors. *Proceedings - 2020 IEEE 5th International Conference on Data Science in Cyberspace, DSC 2020*, 400–406. https://doi.org/10.1109/DSC50466.2020.9194468

Wei, L., Zhang, Y., Yin, D., Shi, Y., Deng, X., & Deng, S. (2021). *Survey on APT Attack Detection in Industrial Cyber-Physical System*. 296–301. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9707300

White, F. E. (1991). Data Fusion Lexicon. *The Data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3*, *15*(0704), 15. http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA529661%5Cnhttp://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA529661

Wikipedia. (2019). *Cyberspace - Wikipedia*. https://en.wikipedia.org/wiki/Cyberspace

Wikipedia. (2022). *Situation Awareness*. https://en.wikipedia.org/wiki/Situation_awareness

Wikipedia. (2023a). *Hopkins statistic*. https://en.wikipedia.org/wiki/Hopkins_statistic

Wikipedia. (2023b). *Longest common subsequence*. https://en.wikipedia.org/wiki/Longest_common_subsequence#:~:text=A longest common subsequence (LCS,positions within the original sequences.

Wikipedia. (2023c). *Silhouette (clustering)*. https://en.wikipedia.org/wiki/Silhouette_(clustering)

Wilkens, F., Ortmann, F., Haas, S., Vallentin, M., & Fischer, M. (2021). Multi-Stage Attack Detection via Kill Chain State Machines. In *CYSARM 2021 - Proceedings of the 3rd Workshop on Cyber-Security Arms Race, co-located with CCS 2021* (Vol. 1, Issue 1). Association for Computing Machinery. https://doi.org/10.1145/3474374.3486918

Wright, K. (2022). *Will the Real Hopkins Statistic Please Stand Up?* https://doi.org/10.32614/RJ-2022-055

Wunder, J., Corporation, M., Markdavidsonnccom, M. D., Jordan, B., & Corp, S. (2017). *TAXII ™ Version 2 . 0*. *July*. http://docs.oasis-open.org/cti/taxii/v2.0/taxii-v2.0.pdf

Wunder, J., Halbardier, A., & Waltermire, D. (2011). *Specification for Asset Identification 1 . 1*. https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7693.pdf

Wynn, J., Whitmore, J., Upton, G., Spriggs, L., McKinnon, D., McInnes, R., Graubart, R., & Clausen, L. (2011). Threat Assessment & Remediation Analysis (TARA). *MITRE Technical Report (MTR110176)*, *October*, 60. https://apps.dtic.mil/sti/pdfs/ADA576473.pdf

Xiong, W., & Lagerström, R. (2019). Threat modeling – A systematic literature review. *Computers and Security*, *84*, 53–69. https://doi.org/10.1016/j.cose.2019.03.010

Xiong, W., Legrand, E., Åberg, O., & Lagerström, R. (2022). Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Software and Systems Modeling*, *21*(1), 157–177. https://doi.org/10.1007/s10270-021-00898-7

Yadav, T., & Mallari, R. A. (2016). *Technical Aspects of Cyber Kill Chain*. https://arxiv.org/pdf/1606.03184.pdf

Yannakoudakis, H. (2018). *Viterbi Algorithm for HMM Decoding Last session : estimating parameters of an HMM*. https://www.cl.cam.ac.uk/teaching/1718/MLRD/slides/slides9.pdf

Ye, N., Zhang, Y., & Borror, C. M. (2004). Robustness of the Markov-Chain Model for Cyber-Attack Detection US Air Force Office of Scientific Research. *IEEE Transactions On Reliability, Vol. 53, No. 1, March 2004*, *53*(1), 116–123. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1282169

Yi, S., Peng, Y., Xiong, Q., Wang, T., Dai, Z., Gao, H., Xu, J., & Wang, J. (2013). *Overview on Attack Graph Generation and Visualization Technology*. https://ieeexplore.ieee.org/abstract/document/6825274

Yu, T., Xin, Y., Zhu, H., Tang, Q., & Chen, Y. (2022). Network Penetration Intrusion Prediction Based on Attention Seq2seq Model. *Security and Communication Networks*, *2022*. https://doi.org/10.1155/2022/6012232

Yu, Z., Wang, J., Tang, B., & Lu, L. (2022). Tactics And Techniques Classification In Cyber Threat Intelligence. *The Computer Journal*, *00*(0). https://doi.org/10.1093/comjnl/bxac048

Zhang, A. F., Li, Z. T., Li, D., & Wang, L. (2007). Discovering novel multistage attack patterns in alert streams. *International Conference on Networking, Architecture, and Storage, NAS 2007*, *Nas*, 115–121. https://doi.org/10.1109/NAS.2007.20

Zhang, R., Huo, Y., Liu, J., & Weng, F. (2017). Constructing APT Attack Scenarios Based on Intrusion Kill Chain and Fuzzy Clustering. *Security and Communication Networks*, *2017*. https://doi.org/10.1155/2017/7536381

Zhu, B., Joseph, A., & Sastry, S. (2011). A taxonomy of cyber attacks on SCADA systems. *Proceedings - 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, IThings/CPSCom 2011*, 380–388. https://doi.org/10.1109/iThings/CPSCom.2011.34

Zhu, Z., & Dumitras, T. (2018). ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. *Proceedings - 3rd IEEE European Symposium on Security and Privacy, EURO S and P 2018*, 458–472. https://doi.org/10.1109/EuroSP.2018.00039

Zibak, A., Sauerwein, C., & Simpson, A. (2021). A success model for cyber threat intelligence

management platforms. *Computers and Security*, *111*, 102466. https://doi.org/10.1016/j.cose.2021.102466

Zibak, A., & Simpson, A. (2019a). *Cyber Threat Information Sharing : Perceived Benefits and Barriers*. https://dl.acm.org/doi/pdf/10.1145/3339252.3340528

Zibak, A., & Simpson, A. (2019b). Towards better understanding of cyber security information sharing. *2019 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cyber SA 2019*. https://doi.org/10.1109/CyberSA.2019.8899697