UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

The Anatomy of Cyber Threats: A Natural Language Approach to Asset-Based Threat Modelling

by

Tom J. Bell
MEng (Electronic Engineering) PGCert MIET CISSP

A thesis for the degree of
Doctor of Philosophy

March 2026

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Doctor of Philosophy

The Anatomy of Cyber Threats: A Natural Language Approach to Asset-Based Threat Modelling

by Tom J. Bell

The pervasiveness and criticality of emerging technologies throughout personal life, business, government and defence, is generating a growing demand for more advanced cyber threat detection, analysis and response capabilities. Threat modelling is a core component of these activities, and asset-based threat modelling in particular constitutes a primary methodological approach for characterising and understanding cyber threats in a broad range of technological and industrial domains. However, existing asset-based threat modelling processes typically exhibit a range of methodological limitations which can significantly constrain the validity of any resultant threat models.

Accordingly, this work develops and presents a generalised Reference Framework for Asset-based Threat Modelling (ReFAThM) intended to guide the development of such threat models, evaluate their completeness, and thus improve their robustness. We also present a novel automated method for characterising the threat landscape using topic modelling. Here, the CWE dataset is used as ground-truth and is pre-processed to synthesise 12 text-based threat attributes for each, using an LLM. These combined attributes constitute the input to topic modelling. Following a cluster merging process, this yields a concise set of 19 threat types, without undermining the breadth and depth of the originating vulnerability dataset. In a subsequent research activity, we utilise these 19 clusters in a classification paradigm to conduct feature importance analysis over the original 12 threat attributes. We quantitatively establish that the 'vulnerability', 'countermeasures', 'detection method', 'technical impact' and 'relevant assets' are the most important threat attributes when characterising a cyber threat. These findings are then employed within an ontology engineering process to develop a 'core threat model' which is suitable to form the basis of more specialised asset-based threat models in a broad range of domains.

Contents

List of Figures

List of Tables

Listings

Declaration of Authorship

I declare that this thesis and the work presented in it is 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. Parts of this work have been published as:
   Chun Man Tsang, Tom Bell, Antonios Gouglidis, and Mo El-Haj. Deciphering Cyber Threats: A Unifying Framework with GPT-3.5, BERTopic and Feature Importance. In *The 1st International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security (NLPAICS), Lancaster, UK. 2024.*, pages 175–185, 2024

Signed:..... Date:.....

Acknowledgements

My foremost thanks and wholehearted praise goes to the God of this universe who created and holds together all things (Colossians 1:16-17). He is my rock and my redeemer, and in his Son the Lord Jesus, has given me spiritual life that I might know Him, be forgiven by Him and enjoy peace with Him. This God upholds the laws of nature, and in Christ - the Word of God - He has revealed Himself as the wellspring of all life, purpose, truth, goodness and love (John 1:1-14). His grace is altogether unmerited and I count it the greatest conceivable joy that I can completely trust him with my future.

It has been a privilege to continue my studies at the University of Southampton within the Cyber Security Research Group. My sincerest gratitude goes to my supervisors, Vladimiro Sassone and Mike Surridge, who provided me with valued advice throughout.

Our dear friends at Christ Church Southampton have provided me with inestimable support, encouragement and prayer. They constantly remind me that the fear of God is the beginning of wisdom. Thank you for challenging me to devote all the blessings and privileges which He has given me to His glory.

I cannot express more fervent gratitude to my parents, who have loved me since before I was born. Thank you for telling me about Jesus, for modelling to me a life of self-sacrifice, and for instructing me in the ways of righteousness.

Finally, words simply cannot describe the extraordinary love that my beloved wife Jenna has shown me throughout these studies. Thank you for encouraging me to take on the challenge, and for your patience with me through the many highs and lows. It has been your support and encouragement that got me through.

Chapter 1

Introduction

As the pervasiveness of emerging technology continues to intensify, the threat landscape presents an ever changing environment for security analysts, data owners and regulators. Attack surfaces are becoming harder to identify and to define, and the prevalence of varying threat actors continues to demand a growing investment of time, resources and management in order to appropriately mitigate.

At the beginning of 2024, there were 5.5 million businesses operating in the UK. As published by the Department for Business Energy and Industrial Strategy, 99.9% of them were small and medium-sized enterprises (SMEs) and 99.1% had fewer than 50 employees. These businesses employed 60% of the UK workforce in 2024. Such patterns are replicated across the European Union (European Commission (2019)) and the United States (U.S Bureau of Labor Statistics (2019)). The UK's Cyber Security Breaches Survey of 2023 notes that, although larger organisations are more likely to experience cyber security breaches, 32% of businesses have knowingly suffered a breach in the previous 12 months, as have 24% of charities (DCMS (2023)). These breaches cause significant damage, such as direct expenses, loss of business, fines and reputational damage. As such, one of the most significant business risks faced by SMEs is the threat posed by possible cyber attacks (Heitzenrater and Simpson (2016)).

The provision and exploitation of accurate, comprehensive and up-to-date cyber threat intelligence (CTI) provides the foundation for a range of cyber risk management processes and tools. Whether it is threat analysis, threat modelling, risk assessment or vulnerability research tasks, cyber security researchers and professionals rely on using CTI in order to understand the threat landscape and seek to respond appropriately to it. However, many of these processes are notably cumbersome, error prone, time-consuming, and liable to quickly becoming out of date.

In view of the inefficiencies in manually analysing and evaluating CTI, researchers within academia and industry have, for some time, been developing and utilising

techniques for automating aspects of the analysis and evaluation of CTI. This has contributed to significant advances across the academic field of cyber security and within industry, by accelerating the process of threat modelling and analysis, improving its accuracy and widening the scope of its applicability to addressing emerging cyber threats.

For instance, in 2014, the UK government introduced its *Cyber Essentials* (CE) scheme which is a framework for helping organisations to establish a basic level of protection from cyber threats, along with a corresponding certification awarded to organisations who have demonstrated compliance. CE was developed in conjunction with the British Standards Institution (BSI), the Information Assurance for Small and Medium Enterprises (IASME) consortium and the Information Security Forum (ISF) (Department for Business Innovation & Skills (2015)). It specifies a set of technical controls which are intended to help such organisations protect themselves from a range of common cyber threats. It is often assumed that CE was designed for SMEs alone, however it was actually intended to be size-agnostic (Chris Ensor (2017)). In spite of this, it does tend to be smaller organisations who have tighter constraints on the availability of financial resources, capacity and expertise to put in place effective cyber controls. This contributes to an increased need for support and guidance regarding the prioritisation and selection of controls which will achieve the greatest marginal improvement to their cyber resilience.

By way of example of one set of mitigations which would be an interesting subject of the kinds of analysis discussed herein, *Cyber Essentials* contains five broadly defined technical controls which organisations should implement (NCSC (2020)):

1. **Secure your internet connection**. Use a firewall with properly configured rules to protect devices, especially if they connect to untrusted Wi-Fi networks. Remove networked services which are non-essential or vulnerable.

2. **Secure your devices and software**. Change default configuration to remove unnecessary services and use strong passwords. Only necessary software, accounts and applications should be active.

3. **Control access to your data and services**. Access to user accounts, devices, software and cloud services should operate on the basis of least privilege. Administration privileges should be reserved for those who need them and should be limited to what is necessary.

4. **Protect from viruses and other malware**. Anti-malware software should be active in order to identify and mitigate malicious files, websites and emails. This may include white-listing, sand-boxing and running regular malware scans.

5. **Keep your devices and software up to date**. Patches should be applied to computers and network devices by updating their software.

The essential problem which CE seeks to address concerns the optimisation of the selection of cyber controls in response to threats, within a resource-constrained environment (such as within the context of the limited budgets of SMEs). When viewed as an optimisation problem in this context, a critical challenge for threat modelling is to accurately model the effectiveness of cyber controls in order to determine the set of controls that would have the most significant marginal impact at reducing cyber-related risk.

Within this context, automating the process of threat modelling could be used to expedite the analysis of cyber countermeasures (such as, for example, validating and iterating on the the control requirements of cyber essentials), with regards to their effectiveness at mitigating threats (i.e. known commodity-level threats derived from sources such as the CVE or CWE databases) which exist for a target IT system (i.e. a network representative of a typical SME). Such a threat modelling process could also contribute to the validity and accuracy of such analysis, and its ability to be re-run regularly as threat information, IT systems and available mitigations evolve. Combined, this could give users greater confidence in the relevance of the controls required by the CE scheme.

While the validation and improvement of the CE controls is one example of the motivation for this research, the scope of its motivation is much wider. There are an extremely broad range of important outcomes of threat modelling, all of which could - in principle - be improved, with a more robust and accurate methodological basis for threat model development.

1.1 Research Problem

A range of techniques have been employed to improve the pace and quality of threat analysis and countermeasure evaluation. A core activity for addressing this challenge is to use one of a variety of cyber threat modelling techniques. Cyber threat modelling is typically conducted by modelling the cyber threat from the perspective of the *software* vulnerabilities, the profile of the *attacker* or by modelling the *assets* in a given system.

Asset-based modelling in particular has been shown to be advantageous in a number of respects (Shostack (2014)), including its capacity for conducting automated reasoning over a threat knowledge base. However the validity of the output of an asset-based threat modelling process is contingent upon the validity of the threat model itself. In particular, the selection, arrangement and configuration of the main concepts which underpin the threat model are critical to justifying the robustness of its capacity to reason in a way that is representative of the underlying dynamics of a real cyber threat (and any mitigations which may be modelled). This is typically done simply by using qualitative techniques, such as being based on the subjective expertise of the threat

model creator. However this method lacks repeatability and makes it harder to justify the design of the threat model.

The asset-based approach to cyber threat modelling is commonly used for characterising a network or IT system to identify threats and opportunities to mitigate them. However, the threat modelling activity is still significantly limited in its utility due to the *robustness* of the threat modelling process; the *coverage* of the threat knowledge base which it utilises, and the *expressiveness* of the threat model at characterising a given threat. It is these three attributes of threat modelling which this thesis seeks to address and improve.

In particular, the research problem addressed within this thesis is concerned with addressing the effectiveness of asset-based cyber threat modelling by focusing on the following key limitations in the existing literature:

- **Robustness:** There is a lack of an overarching framework to guide and facilitate the development of robust asset-based threat models, making it difficult to assess the validity or completeness of a given threat model, or compare its merits with those of other models.
- **Coverage:** There is also no technique to generate a concise and expressive threat knowledge base repeatably, without disproportionately compromising the coverage (i.e. breadth) of the threat landscape, leading to threat models being used to reason using insufficient primary datasets as their ground truth and consequently limiting the validity of their findings.
- **Expressiveness:** There is also a lack of a technique to automatically characterise the attributes of a cyber threat model, which means that most new threat models are designed primarily using qualitative techniques, such as on the basis of the subjective judgement of their originator. Therefore, a robust quantitative technique is needed to identify which fundamental concepts (attributes) relating to cyber threats are most pertinent to modelling the composition and configuration of a cyber threat.

1.2 Research Aim and Questions

As such, the aim of this research is to design and demonstrate the successful use of a set of techniques for improving the robustness, expressiveness, and coverage of the threat modelling process. These techniques should be suitable for combined use as part of the development of a cyber threat model, including one to be used for the analysis of the effectiveness of cyber countermeasures, such as those set out in the *Cyber Essentials* certification.

The research questions which we set out to answer in this thesis are as follows.

1. What activities are necessary to sufficiently characterise a robust and generic asset-based threat modelling process?
2. How can existing techniques in natural language processing (NLP) be used to derive a consolidated cyber threat knowledge base from an unstructured threat database?
3. To what extent can feature importance in classification be used as a quantitative measure of the relative importance of cyber threat attributes for the development of a cyber threat model?

1.3 Research activities and contributions

In order to answer the above-mentioned questions, we have adopted an approach which combines a range of different areas of research within computer science. We begin with the limitations in existing CTI research and datasets, before using a systematic-literature review of various threat modelling techniques. We then leverage a selection of techniques and models within machine learning, and NLP in particular, to characterise a cyber threat knowledge-base and a threat model.

The technique developed in this paper seeks to address these research challenges by first clustering primary cyber threat information into clusters corresponding to normative threat classes. Secondly, by conducting feature importance analysis to identify the relative importance of each threat attribute, we are able to identify the most important concepts which should constitute the composition of a generic threat model for conducting asset-based cyber threat modelling.

Finally, we use ontology engineering to propose a threat model on the basis of the findings of the previous stage. Modelling of any kind is an inherently reductionistic process. However, there is great utility in employing appropriate threat models for research, analysis and risk management purposes, and therefore there are high stakes in the design and execution of an appropriate method for developing them.

This work contains the following contributions, which are of interest to the research community:

1. A review and discussion of the background literature discussing various approaches to asset-based cyber threat modelling, and an identification of some key research gaps.

2. A review and discussion of the principles, methods and evaluation criteria related to ontology engineering in the context of asset-based threat modelling.
3. A novel Reference Framework for Asset-based Threat Modelling (ReFAThM), to guide the development and evaluation of asset-based cyber threat models.
4. A novel method for quantifying the characterisation of a knowledge domain, such as that of cyber security, involving the synthesis of a multi-attribute dataset, topic modelling and cluster merging, followed by classification and feature importance analysis.
5. A concise, broad and expressive taxonomy of cyber threat types, for use within threat modelling and cyber threat intelligence for academic and commercial applications.
6. A quantification of the relative importance of 12 cyber threat attributes to assist in addressing the challenge of threat characterisation.
7. A generalised asset-based 'core' cyber threat model, to form the basis of context-specific threat models in further research or commercial use cases.

1.4 Thesis structure

Chapter 2 provides a review of the peer-reviewed literature on cyber threat intelligence and threat modelling, as well as a critique of the literature and identification of some identified gaps. It further discusses the literature in relation to methodological approaches in the field of ontology engineering, including its relevance to cyber threat modelling.

In Chapter 3 we present the specific research questions which this thesis seeks to address, as well as the research methods which are employed. This chapter concludes with an overall research methodology adopted throughout and justifies why key methodological decisions were made.

In order to begin to understand the key aspects of the research problem we are interested in, Chapter 4 describes the implementation of a systematic literature review which is used to characterise a generic asset-based threat modelling process. This chapter helps to make such a process explicit and provides a framework for practically assisting in the design and development of such threat models. It further enables the significance of subsequent chapters to be understood in this context.

Chapter 5 presents a method for automatically generating a taxonomy of cyber threats produced using a method involving the use of topic modelling, and a bespoke cluster

merging process, to perform hierarchical clustering of human-readable descriptions of cyber weaknesses, as well as the resultant taxonomy itself.

In order to produce a quantitative justification for the selection of threat attributes in a cyber threat model, Chapter 6 outlines a process for measuring the relative importance of individual features of a multidimensional classification paradigm. These features represent threat attributes of a threat model and enable a quantitative measure of an attribute's importance to be determined in a highly repeatable manner. The chapter concludes by presenting a 'core' asset-based threat model based on the results of this evaluation.

Finally, Chapter 7 summarises the main conclusions and contributions of this thesis, as well as outlining some of its identified limitations and options for further work.

Chapter 2

Background and Related Work

This research addresses many of the existing challenges associated with cyber threat modelling, in order to enable the research community to perform threat modelling in a methodologically robust manner and using appropriate cyber threat information. The research problem is summarised in Section 1.1 and details the main challenges associated with asset-based cyber threat modelling. Accordingly, we aim to develop an approach for overcoming these challenges through the exploitation of a range of techniques found in the relevant literature, and by supplementing those techniques with our own novel contributions.

Therefore, this chapter reviews the relevant literature to this research problem and our proposed solutions. This includes discussing the wider cyber security context of businesses, semantic modelling and existing approaches and techniques for asset-based threat modelling, and a discussion of cyber threat intelligence (CTI).

We also review the domain of ontology engineering in order to provide a rigorous foundation for an ontology engineering methodology used to develop a semantic threat model for countermeasure analysis. This includes a description of the various approaches to designing ontologies as well as how they relate to taxonomies. Following this is an overview of the general methodologies available for ontology design and the techniques used for evaluating the quality of an ontology.

2.1 Cyber Security for Small Businesses

Small to Medium-sized Enterprises (SMEs) face a range of business risks on a day to day basis. One study mentions six types of risk for SMEs as being interest rates, technological risks, raw material prices, supply chain risks, growth risks and workforce risks (Falkner and Hiebl (2015)). Among the technological risks, those related to cyber security are particularly notable in their importance and there are a range of important

perspectives on the unique cyber security context faced by SMEs. A systematic review has identified five major aspects of cyber security risk management among SMEs (Alahmari and Duncan (2020)). The following discussion is structured on the basis of these aspects.

Knowledge of cyber security *threats* is essential to enabling SMEs to understand their cyber security context, however SMEs often under-appreciate their exposure to threats, such as data breaches, data destruction and denial of service. Renaud and Weir (2016) identify a lack of skills, resources and perseverance as contributing to a failure to implement appropriate countermeasures to mitigate these threats. This has been shown to manifest itself in unknown outsourcing of responsibility and the use of old security methods (Barlette et al. (2017)).

Further, the *behaviours* exhibited by employees within SMEs can lead to an increased threat profile, such as failing to consistently apply security policies, guidelines and company rules Barlette et al. (2017). While training can help to induce the right behaviour, there is a notabe attenuation in awareness of cyber security best practice after the training is undertaken (Gundu (2019)). Indeed, rather than knowledge being the greatest indicator of cyber security awareness *per se*, Kaur and Mustafa (2013) show that it is the behaviour and attitude of employees which has a more significant relationship with awareness.

While there is a growing professionalisation of cyber security, SMEs appear not to be responding as quickly as larger companies to expectations of professional cyber security *practices* (Kabanda et al. (2018)) leading to unconscious participation in risky practices. While procuring cloud-based services may offer security advantages, Osborn and Simpson (2018) show that it does not replace the need for the adoption of good practices. In addition, while training and education can enhance knowledge, it is another matter as to whether this is translated effectively into the adoption of secure practices (Gundu (2019)).

SMEs often have a low level of *awareness* of cyber-criminality by virtue of a lack of knowledge of the types of cyber attacks they may be vulnerable to (Osborn and Simpson (2018)), and it is this awareness that is crucial to introducing appropriate cyber security measures (Kabanda et al. (2018)). Bada et al. (2019) show that SMEs must be aware of the consequences of cyber security as this awareness can motivate them to adopt appropriate behaviours. Accordingly, it may be possible to achieve acceptable levels of cyber risk by creating a suitable awareness programme (Gundu (2019).

The last perspective considered is the role that cyber security *decision-making* has on the effectiveness of risk management. Bayaga et al. (2017) discuss how critical the management of a company is at mitigating cyber threats by virtue of their decision-making responsibilities. In most SMEs, it is generally executives who decide the cyber security strategy of their company (Kabanda et al. (2018)) and therefore the role of expert advice

at that level is vital for influencing the overall awareness of cyber security across the company (Barlette et al. (2017)). Furthermore, since cloud service providers play an important role in ensuring users have access to usable security controls (Parkin et al. (2016)), decision-makers must be aware of what level of controls are offered.

2.1.1 Understanding SME network architectures

Asset models to be used within a semantic reasoning process are often constructed with reference to a system archetype. These are models of an IT network which can be used to represent the components of a typical network and how they interact with each other. This makes it possible to more easily produce an asset model to represent these components.

One approach to developing a network architecture has been to ask participants who run small businesses to provide a realistic description of an IT system in such an organisation. However Caralli et al. (2007) suggest that at least two participants should be involved in the process of determining a representative architecture at a given time, in order to produce meaningful results.

Such et al. (2015) used a survey of SMEs in order to produce a set of case study network architectures. Their case studies included representative networks of a financial organisation, a specialist group, a web development organisation and a hotel service. They also developed a 'typical network' for an SME (illustrated in Figure 2.1) using an incentivised survey method, which displays features of each of their case study networks. These network architectures were then used to assess the performance of Cyber Essentials at mitigating a set of known vulnerabilities.

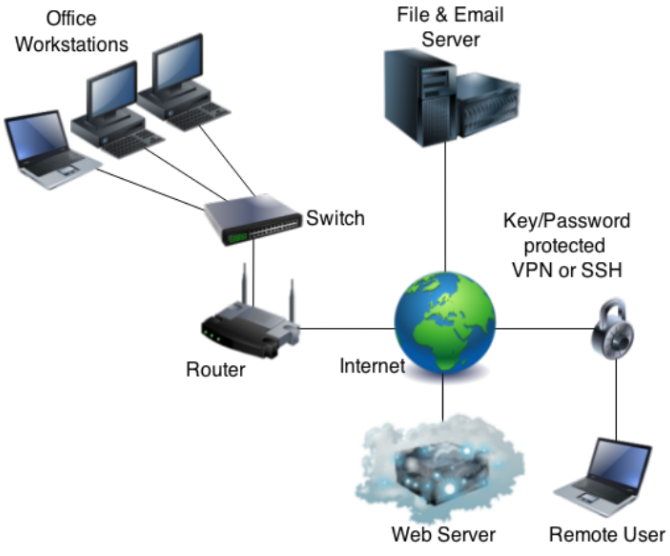


FIGURE 2.1: A typical network as adapted from a set of incentivised survey-response results (Such et al. (2015)).

Network architectures can also be represented using a model based on multiple depth layers (Panaousis et al. (2014)), such as, a *Demilitarised Zone (DMZ)* depth layer representing a layer of assets directly accessible via the internet, a *middle-ware* depth layer representing enterprise assets and a proxy (such as a VPN), and a *private network* depth layer representing data assets which are highly sensitive and only accessible via the proxy in the middle-ware. In this case, each layer is separated by a set of network security capabilities, such as firewalls, intrusion detection and network access control assets. Network models comprising DMZs are widely recognised and have been used in other studies, such as in risk-awareness and educational scenarios (Hart et al. (2020)).

Parkin et al. (2016) developed a set of SME system archetypes using UK-based statistics on the number of SMEs which operate from a single site and from the owner's home, as well as those which use shared offices, or are mobile. This enables them to construct a set of archetypal systems which vary on the basis of the SMEs size, network design and the type of daily interactions with the IT system which might be a source of weakness. These system archetypes are used as an input to a model of security effort investment.

The approach from Caralli et al. (2007) is used by Osborn and Simpson (2018) to assess the effect of constraints faced by SMEs on their risk mitigation. The system architecture from their scenario is illustrated in Figure 2.2 and shows how users can interact with the operations of the business from a range of locations, can use BYOD, and can interact with cloud services and external parties. The previous work in Osborn (2014) uses a similar questionnaire-based method to produce a set of archetypal network models for SMEs of different sizes, for both those with and without their own office.

2.2 Cyber threat intelligence

The field of Cyber Threat Intelligence (CTI) has emerged in recent years as referring to the task of gathering, analysing and using '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' (McMillan (2013)). Sharing and using CTI can help to improve the resilience of organisations by enabling them to more accurately understand the threats they may be exposed to and providing the decision support to proactively mitigate the greatest threats (Wagner et al. (2019)).

2.2.1 Common definitions

In order to maintain consistency and integrability, it is important to establish definitions for important kinds of information relating to threat intelligence (Bromander et al.

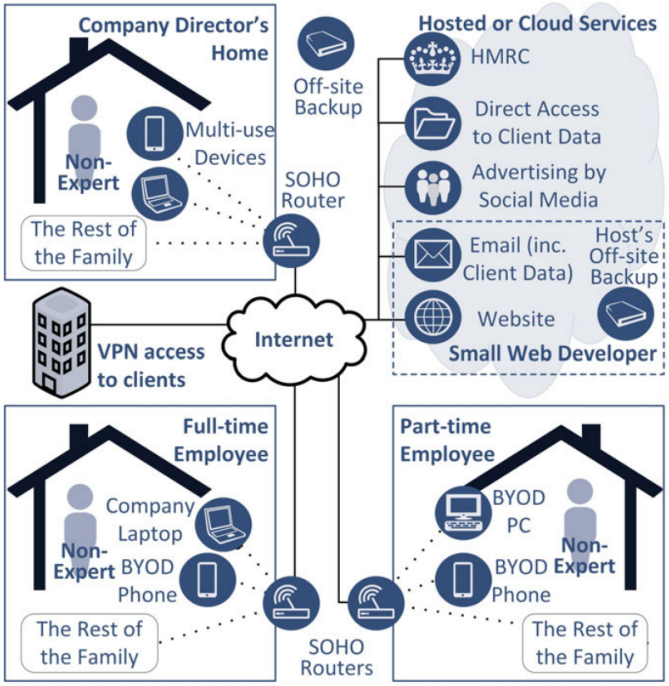


FIGURE 2.2: A scenario architecture of a small business with various employees using different means of interacting with the business (Osborn and Simpson (2018)).

(2020)). In view of the importance of a suitable knowledge base for performing automated threat modelling, we will base our definitions on those provided in Sauerwein et al. (2019) which are consistent with their taxonomy of data sources. These definitions are consistent with those set out in the ISO 27000[1] standard:

- *Vulnerability*: A weakness of an asset which might be exploited by a threat.
- *Threat*: The potential cause of an unwanted incident.
- *Countermeasure*: Any administrative, managerial, technical or legal course of action used to mitigate an information security risk.
- *Attack*: Any unauthorised attempt to maliciously exploit a vulnerability to access, alter or destroy an asset.
- *Risk*: The consequences and likelihood of a potential event, such as an attack.
- *Asset*: Any object or characteristic that has value to an organisation.

2.2.2 Types of CTI information

There may be considered to be two abstract types of cyber threat intelligence: informal sources and formal sources. Informal sources includes online forums, specific Twitter

[1]ISO/IEC 27000: https://www.iso.org/standard/73906.html

accounts, Stack Overflow, blogs, news reports and other websites. Informal sources can offer many benefits and as such they have been employed along with natural language processing (NLP) techniques to extract actionable CTI. However, for our present purposes, informal data sources generally fail to achieve the coverage, expressiveness and standardised representation necessary to provide a rigorous scientific basis for use when conducting threat modelling and analysis.

Formal sources generally refer to enumerations of structured (or partially structured) threat information, with a consistent representation and quality. Each of these data sources are specific kinds of CTI, each of which complement each other and are suitable for different types of threat modelling.

The essential nature of, and relationships between, the selected data sources have been identified as illustrated in Figure 2.3. The rest of this section provides a discussion of these formal data sources.

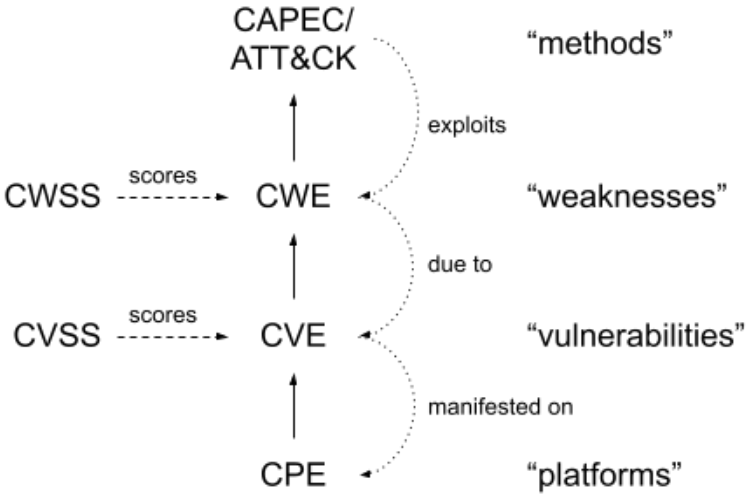


FIGURE 2.3: An illustration of the content of, and relationships between, enumerations and scoring systems.

2.2.3 Enumerations

In order to characterise threats within a semantic model, it is necessary to have access to a suitable catalogue of threats which can be used to populate the model before any analysis may take place. Mavroeidis and Bromander (2017) identify three types of taxonomy which may be relevant to the design of a threat catalogue as being *enumerations*, *scoring systems* and *sharing standards*.

Mitre maintains the Common Vulnerabilities and Exposures (CVE) (MITRE (d)) database which provides a concise human-readable description of publically-known software vulnerabilities and assigns a unique identification number to each one.

Mitre's Common Weakness Enumeration (CWE) dictionary (MITRE (a)) is a database of common weaknesses, largely derived from the CVE database, which defines software and hardware weaknesses in order to better understand types of vulnerabilities. Most CWE entries contain suggestions of appropriate countermeasures to mitigate each weakness.

The National Vulnerability Database (NVD) (NIST (b)) is a database managed by NIST which comprises a repository of standards-based vulnerability management data. It is represented using the Security Content Automation Protocol (SCAP) (NIST (c)) and is intended to facilitate the automation of vulnerability management by including security checklist information, software vulnerabilities and impact metrics. NVD is built on top of CVE information, but also incorporates Common Platform Enumeration (CPE) (MITRE (c)) and CWE information into each NVD entry.

The Common Platform Enumeration (CPE) (MITRE (c)) is Mitre's machine-readable standard for assigning and encoding names for software and hardware IT products.

Mitre also provides a collection of the most common attack patterns used in cyber attacks in order to exploit known weaknesses. This collection is known as the Common Attack Patterns Enumeration and Classification (CAPEC) (MITRE (b)) and includes a description of each attack pattern, its relation to other attacks and the consequences of successful exploitation. Similarly to CWE, most CAPEC entries also suggest possible countermeasures to mitigate their attack patterns.

In addition, Mitre maintains the ATT&CK (MITRE (h)) taxonomy which provides a collection of known threat actors, and their techniques and tactics, based on real-world observations, and is used for a range of existing threat modelling techniques. Whereas CAPEC is a comprehensive collection of techniques which covers the whole life-cycle of an attack, ATT&CK is primarily focussed on post-compromise techniques used to achieve a particular objective and has a focus on securing networks. The techniques specified in ATT&CK may use the attack patterns in CAPEC.

The Centre for Internet Security (CIS) maintains a collection of security controls called the Critical Security Controls (CSCs)[2] as best practice recommendations for providing improved cyber security. They comprise 20 controls, each with a set of sub-controls which reflect various stages of the attack life-cycle. While there is no direct correspondence between the CIS controls and the controls of Cyber Essentials, they are represented at a similar level of abstraction and so they are appropriate for being mapped to the Cyber Essentials controls in cases where Cyber Essentials covers them. The CIS controls are also suitable for implementation in the cloud across multiple service models and deployment models, and the CIS provides documentation to assist that process[3].

[2]https://www.cisecurity.org/controls/
[3]https://www.cisecurity.org/white-papers/cis-controls-cloud-companion-guide/

Another important enumeration is that provided by the VERIS[4] schema. VERIS is a vocabulary for recording cyber incidents comprising a set of schema enumerations[5] for representing security incidents in a structured manner. It includes enumerations to classify the threat landscape, using the following primary categories:

1. **Actors**, describing those who are responsible for an incident,
2. **Actions**, corresponding to ways in which methods can be used to affect assets, such as malware, hacking, misuse or environmental actions,
3. **Assets**, representing different kinds of devices which could be affected by an incident, and
4. **Attributes**, describing how assets are affected.

By use of the VERIS Common Attack Framework (VCAF), VERIS can be used to extend ATT&CK and provide a conceptual mapping from VERIS to ATT&CK in order to cover all possible threat actions present in VERIS. There is no structured enumeration of countermeasures as part of the VERIS framework *per se*, however VERIS do provide a mapping from each of the CIS's CSCs to VERIS[6] in order to make consistent control decisions when using the VERIS framework.

While the concepts of TTPs (Tactics, Techniques and Procedures) and attack patterns are well-defined concepts in formal CTI models (Stillions (2014)) they may helpfully be summarised, for the purposes of understanding the general utility in using CAPEC and ATT&CK, as being enumerations of attack 'methods', as shown in Figure 2.3.

On the basis of this analysis, the relevant criteria for the selection of a cyber threat enumeration to be used to build a knowledge base is whether it attains a sufficient degree of *coverage* of the distribution of threat information. This is necessary to harness the power of semantic modelling to ensure that the analysis is comprehensive and is not subject to any type of relevant threat being overlooked. In particular, it should have suitable coverage of the relevant concepts of the CTI model (Mavroeidis and Bromander (2017)). In addition, they should be suitably structured and expressive in the way their content is characterised, and be suitable for integration into a semantic threat model.

2.2.4 Scoring systems

NIST also manages the Common Vulnerability Scoring System (CVSS) (NIST (a)) as part of the NVD. CVSS is a measurement standard for scoring vulnerabilities according to their overall severity, including likelihood and potential impact. As shown in

[4] Vocabulary for Event Recording and Incident Sharing
[5] http://veriscommunity.net/enums.html
[6] https://github.com/vz-risk/veris/tree/master/bin

Figure 2.3, each CVE entry (and therefore each NVD entry also) is assigned a CVSS score which provides a standard metric for prioritising the implementation of countermeasures based upon the overall risk associated with CVE vulnerabilities.

Mitre's Common Weakness Scoring System (CWSS) (MITRE (f)) scores the common weaknesses enumerated by CWE using 18 different factors. CWSS can also be used with Mitre's Common Weakness Risk Analysis Framework (CWRAF) (MITRE (e)) in order to identify and analyse the relationship between CWE entries and the technologies deployed in particular business environments.

2.2.5 Sharing standards

Sauerwein et al. (2017) concludes that the most widely used standard for sharing structured CTI is the Structured Threat Information eXpression (STIX) (OASIS CTI (a)). Whilst comparing it to IODEF, VERIS and XARF, Menges and Pernul (2018) reveals that STIX is also the most comprehensive and practicable reporting format. STIX originally evolved throughout discussions among CTI experts regarding the development of a standardised representation for cyber threat indicators (Barnum (2014)). It provides coverage of a broad range of threat information entities and is designed to be a highly expressive, flexible, extensible, automatable, and human-readable representation language for CTI. Its high-level structure contains a range of constructs which have certain attributes and relations to other constructs, such as vulnerabilities, indicators, TTPs, threat actors and exploit targets. With STIX 2.0, previously-named CybOX (Cyber Observable eXpression) *Observables* have been integrated with STIX and are now called STIX Cyber Observables. STIX has also been simplified somewhat but is still intended to cover all the basic requirements for CTI sharing for a range of business and academic needs. CAPEC entries can be integrated into the STIX framework using the CAPEC schema extension.

Malware Attribute Enumeration and Characterization (MAEC) (MITRE (g)) is a sharing language specifically designed for sharing malware information by representing it in terms of its artefacts, attack patterns and behaviours, and is designed to interface seamlessly with STIX (such as with the STIX *Observables* construct) or as a standalone framework.

In parallel to STIX and MAEC, Mitre have also developed the Trusted Automated eXchange of Indicator Information (TAXII) (OASIS CTI (b)) which is an application protocol for establishing CTI exchanges to enable threat information to be shared between organisations independently of any products used to process such threat information. The standards defined by TAXII can be used within automated threat intelligence sharing infrastructure to define how that intelligence is relayed. TAXII can be used in conjunction with STIX which defines the structure of the information which is relayed.

While the majority of enumerations are represented syntactically in XML[7] or JSON[8] formats by default, Asgarli and Burger (2016) verify that semantic exchange formats such as STIX can be used instead of XML-based formats without any feature loss, while providing additional benefits. They also show that STIX objects can be mapped to RDF[9]/OWL[10] objects to facilitate semantic reasoning.

The majority of the sharing standards and enumerations referred to have been represented as RDF/OWL constructs in an open-source GitHub repository[11].

2.2.6 Using CTI for creating a knowledge base

Sauerwein et al. (2019) used a triangulation study to assess public information security data sources. They developed a taxonomy for classifying CTI data sources which characterises data sources along six dimensions: type of information, integrability, timeliness, originality, type of source, and trustworthiness. These dimensions approximately align with the five criteria of actionable CTI as defined by Pawliński et al. (2015). They then score 68 different data sources on the basis of these six criteria to aid the selection of CTI data sources for different purposes.

The same study involved an exploratory survey to elicit the types of data sources used as input to information security risk management processes. These sources included: news websites, expert blogs, security product vendor websites, vulnerability databases, mailing lists, social networks, streaming portals and forums.

Use of a knowledge base within a semantic model places certain requirements on data sources used. The most relevant dimensions are, the type of information provided, its trustworthiness and its integrability. On the basis of the classification results of Sauerwein et al. (2019), social network data provides the most comprehensive coverage of different types of information (including both vulnerabilities and countermeasures), however none of their analysed social media data sources used a structured format and therefore lacked suitable integrability. In contrast, 33% of assessed vulnerability databases used a structured data format and 22% contained both vulnerabilities and countermeasures, including both the CWE and CAPEC data sources.

A primary challenge with building a semantic knowledge base for cyber security is that most security data is unstructured (Sauerwein et al. (2019)) making it difficult to derive actionable CTI from it.

[7] XML: Extensible Markup Language, http://www.w3.org/XML/
[8] JSON: JavaScript Object Notation, https://www.json.org/json-en.html
[9] RDF: Resource Description Framework, http://www.w3.org/RDF/
[10] OWL: Web Ontology Language, http://www.w3.org/2001/sw/wiki/OWL
[11] https://github.com/daedafusion/cyber-ontology

Although the CVE database is unstructured, machine learning and natural language processing can be used to classify CVE entries on the basis of their corresponding CWE entry, and to convert them into machine interpretable formats (Schaad and Binder (2020)). This can then be used to map the constantly-updated CVE database with threats which can then be associated with individual architectural elements (such as DFD elements) of a system model using the threat classes in the STRIDE methodology.

2.3 Threat modelling

Threat modelling refers to the process of characterising cyber threats (such as through an ontology or other semantic representation). It is typically conducted in order to facilitate the identification and analysis of threats. Threat modelling may subsequently involve the analysis and selection of countermeasures or another appropriate course of action in response to the identified threats. *Semantic modelling* refers to the structured organisation of information in order to enable information to be shared, related to each other and used to facilitate reasoning (Aviad and Węcel (2019)).

By their very nature, cyber threats manifest themselves from within the accumulation of deviations from the intended operation of technological systems. In other words, they operate in a very loosely defined space, since the ways in which threats emerge are constantly evolving as new capabilities are introduced into the marketplace. This renders the process of threat modelling to be notoriously challenging, since it seeks to represent an essentially unstructured domain according to a formal structure. However, there is still considerable utility in pursuing this enterprise.

2.3.1 Ontologies

While sharing an analogous set of characteristics (Hadzic et al. (2009), the field of ontolgies within computer science is different to that in philosophy (Corcho et al. (2003)). Ontologies are a type of semantic representation which use a formal structure to represent meaning. An *ontology* has been defined as 'a specification of a conceptualisation' (Gruber (1995)). They are specified by defining the essential nature of the concepts which relate to a particular knowledge domain and the relationships between those concepts.

Although Gruber (1995) formulated this definition, he specifically refers to the idea of a 'conceptualisation' set out by Genesereth and Nilsson (1987) as resembling a particular configuration of a set of entities with respect to each other at a given moment. However, Guarino (1997) suggests that this isn't an appropriate definition and instead provides a formal definition of a conceptualisation as being an expression of 'the intended

meaning of each symbol independently of the particular situation at hand', which can be expressed, for example, in the form of a set of rules (i.e. axioms) which constrain the concepts (Franz Baader et al. (2007)). Accordingly, an ontology corresponds to a particular logical theory.

This definition highlights the distinction between the representation of concepts within an *ontology* and a *knowledge base* to which that ontology corresponds. Guarino (1997) describes that the purpose of an ontology is to describe facts that are *always true* to a particular community of users, by virtue of an agreed-upon vocabulary, however 'a knowledge base may additionally describe facts and assertions...related to a particular epistemic state'. An ontology may therefore be considered analogous to a template, and a knowledge base to a set of instantiations of the ontology which represents a particular state. This is similar to the relationship between a Class and an Object in object-oriented programming (Dillon and Tan (1993)). Accordingly, an ontology contains a clear distinction between the schema (i.e. the ontology itself) and instances of the schema (Gaševic et al. (2009)).

Further, an ontology serves a different purpose than a knowledge base (Maedche and Staab (2001)). Whereas a knowledge base includes the actual knowledge needed to infer answers, an ontology merely specifies the concepts needed to understand that domain. An ontology together with a set of instances of classes constitutes a knowledge base (Noy and Mcguinness (2001)) which can be used to perform machine reasoning in order to determine answers to various problems which the knowledge base can address (Hadzic et al. (2009)).

Regarding its use with respect to ontologies, Hadzic et al. (2009) defines a *concept* as a unit of thought and a *term* as a lexical representation of a concept. Gruber (1993) notes the importance of establishing ontological commitment by agreeing on the concepts and relationships within an ontology. This includes a commitment to use the shared vocabulary coherently and consistently, as well as the semantics (axioms, rules and constraints) of each concept.

Whereas taxonomies merely provide the naming and hierarchical classification of entities, ontologies additionally specify the semantic relationship between those entities. A lightweight ontology may only consist of the concepts, its properties and their relationships, however they are particularly powerful for performing machine reasoning when they additionally encode axioms and constraints into how the concepts relate to each other (Corcho et al. (2003)). The use of ontologies for representing knowledge can enable improved communication, reusability and organisation of knowledge, as well as to facilitate computational inference from within an integrated knowledge base (Gruninger and Lee (2002); Fernández-Breis and Martínez-Béjar (2002)).

Donini et al. (1997) presents the basic ontology reasoning procedures as being the following:

- *Consistency checking* to ensure that there are no contradictions in the ontology.
- *Concept satisfiability* to determine whether it is possible to interpret a knowledge base using the ontology concept.
- *Concept subsumption* to determine relationships between classes and subclasses within a class hierarchy.
- *Instance checking* to determine the class type of an instance from a knowledge base.
- *Conjunctive query answering* to determine the answer to a query (such as a query defined using SPARQL).

While ontologies have a range of uses (including classification and evaluation) when used as a Semantic Web technology, they are also suitable for being used for performing reasoning. This refers to the process of inferring facts from within a knowledge base which have not been explicitly stated.

2.3.2 Ontology types

Guarino (1997) classifies ontologies according to four different types as illustrated in Figure 2.4, each with a different level of dependence on a particular task or point of view. The arrows in that diagram represent the direction of specialisation (i.e. they point to the more general ontology type). These include the following types:

- **Top-level ontologies** represent very general concepts which are independent of a particular problem or domain, such as *object*, *event* or *action*.
- **Domain ontologies** describe the vocabulary of a generic domain, such as *cyber security*.
- **Task ontologies** describe the vocabulary of a generic task or activity, such as *threat modelling*.
- **Application ontologies** describe concepts which are specialisations of both domain and task ontology concepts. They are often roles performed by domain entities, such as a *countermeasure*.

The specific methodological approaches behind the engineering of particular ontologies is discussed later in this Chapter in Section 2.5.

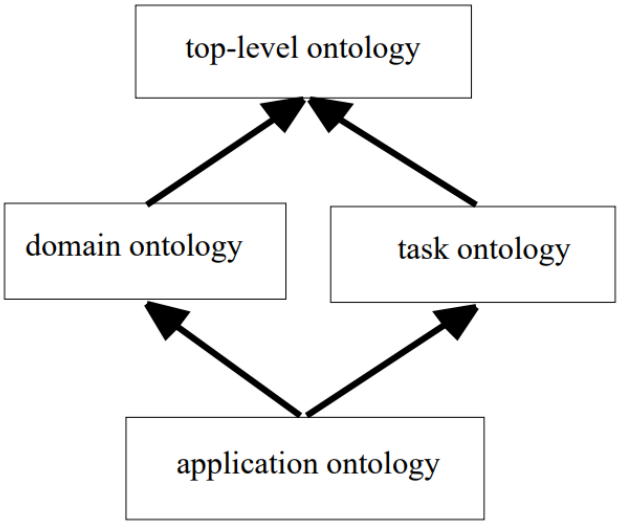


FIGURE 2.4: The kinds of ontologies as specified Guarino (1997), representing the levels of abstraction which may be represented by an ontology.

2.3.3 Threat modelling approaches

Asset-based models are those which focus on the assets which an attacker targets or which are being defended (things of value). Such techniques are particularly helpful in situations where the specific exploit sequences are not well understood. A range of frameworks exist for providing asset-based risk analysis, such as *Trike* (Paul Saitta et al. (2005)) and OCTAVE *Allegro* (Caralli et al. (2007)), however they both rely on predefined catalogues of potential threats. While it may seem intuitive to focus on specific assets of value (targets), asset-based models offer no clear route to inferring what will go wrong with those assets (Shostack (2014)). As such, asset-based methods typically rely on expert understanding of a given IT system and its threat environment. Without this, the threat analysis may be limited in its coverage. Accordingly, asset-based threat modelling techniques benefit from access to broad, expressive and up-to-date catalogues of enumerated threats. This would be necessary in order to make asset-based threat modelling accessible to smaller organisations through automated threat modelling.

Attacker-based models differ in that they consider the agency of a potential threat, and the TTPs which that attacker may seek to employ in order to exploit vulnerabilities. Attack trees, such as *SecurTree* (Ingoldsby (2009)), are one type of such attacker-based model created in to identify attack goals in which threats are modelled by a tree of events which culminate in an exploitation (B. Schneier (1999)). These techniques can be used to identify countermeasures which can interrupt a sequence of exploit events and can also help security analysts to characterise or predict attacker behaviour (Peng and

Zhao (2007)). While such techniques are very powerful, particularly when used alongside databases of attack patterns such as CAPEC (MITRE (b)) or the ATT&CK matrix (MITRE (h)), these methods tend to be vulnerable to analysts failing to identify attackers and the vulnerabilities they could exploit. Further, the TTPs and the personas of attackers are usually not enough to reliably predict what an attacker will do (Shostack (2014)).

Software-based threat modelling focuses on the vulnerabilities and weaknesses in software which may present threats to the system of which that software is a part (Swiderski and Snyder (2004)). Such vulnerabilities and weaknesses may, for example, include those described in the CVE database (MITRE (d)) or the CWE database (MITRE (a)). Software-based approaches are supported by a range of automated tools and other techniques which can be used to identify and prioritise common vulnerabilities. For example, Microsoft's STRIDE approach is provided as an elicitation technique for finding vulnerabilities in software (Khan et al. (2017)) based on a structured analysis of a system typically represented in the form of a data flow diagram (DFD). Other techniques, such as SQUARE (Mead (2005)) are used to enable development teams to integrate security analysis into their production life-cycle through defining secure software requirements. Likewise, UMLSec provides an extension to UML for integrating security information into the software design phase (Jürjens (2002)). Software-based methods are generally the most powerful for performing threat analysis as they are easy to understand, are highly expressive and can be easily represented visually (Shostack (2014)). By virtue of this, they provide considerable scope for automation. Their primary weakness, however, is that they fail to consider non-software-related threats, such as human factors and hardware vulnerabilities.

While there are recognised differences between each of these broad approaches, there is also a degree of overlap. Further, by its very nature as being a generic representation of knowledge, semantic modelling can be used across each of these approaches to modelling cyber threats, or may represent a hybrid approach of more than one of them. For example, many cyber threat ontologies discussed in this chapter constitute a hybrid of asset and software-based threat modelling.

2.3.4 Modelling cyber threat intelligence

As the variety of threat intelligence data changes and the maturity of the CTI sharing ecosystem continues to evolve, this has been accompanied by a development in the models used to represent and evaluate CTI.

In 2013, Caltagirone et al. (2013) published their report introducing The Diamond Model of Intrusion Analysis which has become an important model for understanding the main features of threat events and is complimented by a formal method for conducting

intrusions analysis. This model considers an *event* to be a composition of an *adversary*, *capability*, *infrastructure* and a *victim*. According to the Diamond Model, when an event takes place, the adversary exploits a capability in order to manipulate the infrastructure of the victim.

An entire campaign may comprise a plurality of such events in an activity thread and each event may be grouped together into activity groups to be used to characterise the threat. The authors recognise that their model does not constitute a new ontology or taxonomy, however they note that it can form the basis of a variety of ontologies, as was recognised by Obrst et al. (2012).

Primarily motivated by the emergence of the threat class known as the 'Advanced Persistent Threat' (APT), Hutchins et al. (2011) of Lockheed Martin developed the Cyber Kill Chain model, as illustrated in Figure 2.5, to represent an intrusion consisting of seven distinct phases: *reconnaissance*, *weaponisation*, *delivery*, *exploitation*, *installation*, *command and control (C2)*, and *actions on objectives*. The model is referred to as a chain, since if any one phase of the chain failed, it would interrupt the entire process of the attack. The defender can use the model to understand how to move detection and mitigation controls to earlier phases of the kill chain.

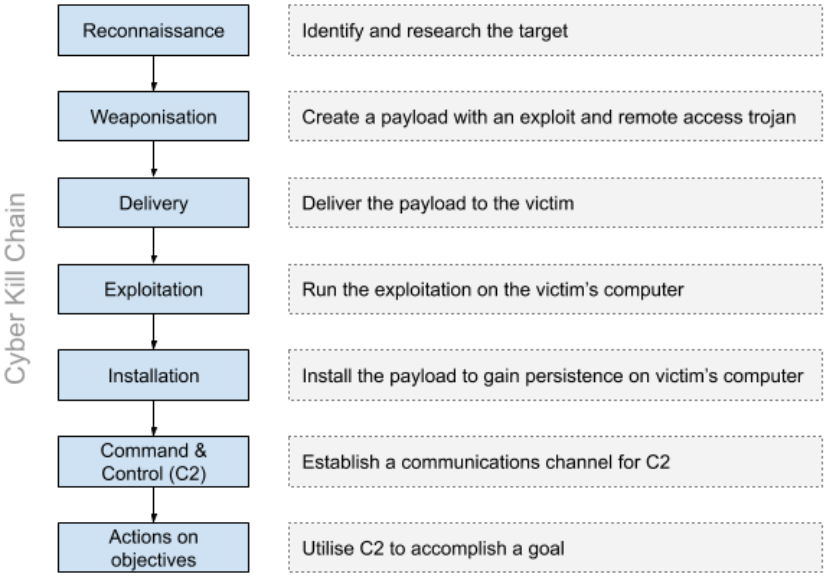


FIGURE 2.5: The Cyber Kill Chain model.

Hutchins et al. (2011) further set out how the basis of intelligence-driven security can be provided by use of the kill chain model, as well as the mapping of kill chain indicators to defender courses of action. They also discuss the importance of identifying patterns that link intrusions (or events, in the language of the Diamond Model) into broader campaigns, and recognise the iterative nature of intelligence gathering.

It can also be helpful to assess CTI in terms of its maturity as suggested by Stillions (2014) who proposed the Detection Maturity Level (DML) model. This model is a hierarchical model of 9 levels of maturity for threat intelligence, whereby lower levels are more specific and higher levels are more abstract. The lower levels indicate evidence left after an attack (*unknown*, *atomic indicators* and *network artefacts*). The middle levels model how the compromise took place (*tools*, *procedures*, *techniques* and *tactics*) and the higher levels refer to what the attacker is seeking to accomplish (*strategy* and *goals*). Bromander et al. (2016) have extended this model to provide the additional level of *identity*, whether an individual, organisation or state actor.

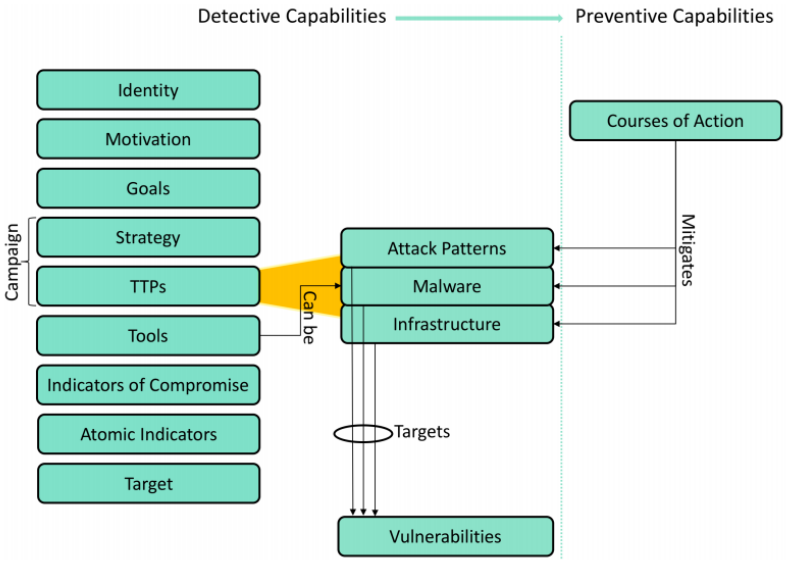


FIGURE 2.6: The Cyber Threat Intelligence (CTI) model (Mavroeidis and Bromander (2017)).

While the DML model facilitated the analysis of detection maturity, Mavroeidis and Bromander (2017) proposed the Cyber Threat Intelligence (CTI) model. The CTI model incorporates the hierarchical DML model from Stillions (2014), but extends it to include the capabilities necessary for the prevention of attacks, as well as for detection. As illustrated in Figure 2.6, it represents the way in which TTPs and tools use attack patterns, malware and infrastructure to target vulnerabilities, but can be mitigated by courses of action. There is no existing ontology for use within CTI which cover all abstraction levels of the CTI model, nor any model which is sufficient for capturing domain expertise in a structured and constrained way to facilitate reasoning (Mavroeidis and Bromander (2017)).

2.4 Semantic approaches to threat modelling

Threat models and CTI are primarily concerned with intrusion detection and response (Bromander et al. (2016)). Of these, use of CTI for intrusion response is at a very early stage of development, largely due to there being no standardised methodology for the validation and comparison of such methodologies (Montemaggio et al. (2020)). However, in addition to detection and response, CTI is also concerned with prevention. While there are a range of semantic models to aid the process of threat analysis, only around 20% of them even support the concept of countermeasures, let alone are primarily concerned with modelling them (Silva and Rafael (2017)).

The use of semantic modelling for automated threat detection and prevention offers promising advantages at multiple levels of abstraction, however the reuse of related standards and taxonomies is necessary in order to control the complexity of such an undertaking (Bromander et al. (2016)). Landwehr et al. (1994) states that taxonomies of threats are particularly useful when they classify threats in terms that correspond to potential defences.

This section explores some of those models which integrate countermeasures into their ontologies to understand their purpose, effectiveness and methodological limitations.

2.4.1 Semantic threat models

One of the earliest ontologies for information security was proposed by Herzog et al. (2007) which represents some general top-level security concepts, such as assets, threats, vulnerabilities and countermeasures, as well as a range of specific concepts. As it is a top-level ontology, it can be applied to many applications and contexts. Fenz and Ekelhart (2009) proposed an extended ontology which additionally includes the concept of an organisation, as well as providing additional concepts which relate to threats. As with many ontologies, 'they struggle to define relationships between unambiguous concepts, such as the distinction between threats and vulnerabilies' (Mavroeidis and Bromander (2017)). It has been further demonstrated by Fenz (2010) that the ontology from Fenz and Ekelhart (2009) can be used to automatically generate IT-security metrics, such as the compliance and quality of control implementations, with respect to various security standards.

As illustrated in Figure 2.7, the ontology proposed by Fenz and Ekelhart (2009) has been simplified by Surridge et al. (2013) in order to reduce the number of assertions that have to be instantiated during run-time analysis of a system. This enabled them to more effectively analyse the model to identify vulnerabilities and diagnose threats. This model was further advanced by Gol Mohammadi et al. (2014) to develop an ontology for run-time trustworthiness as shown in Figure 2.8.

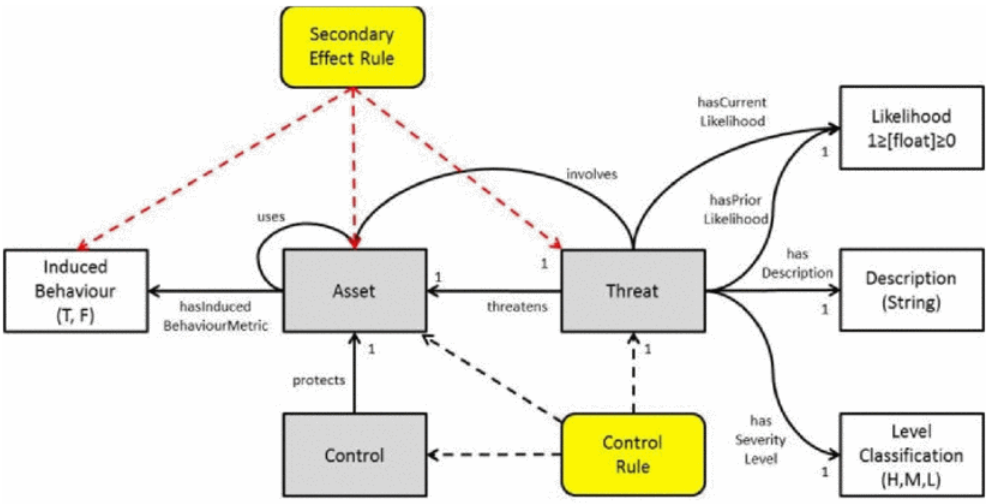


FIGURE 2.7: The core morel classes and SWRL rules of the model used in the SERSCIS project (Surridge et al. (2012, 2013)).

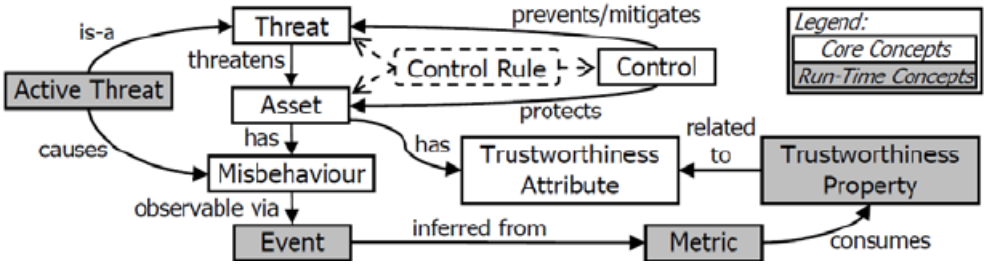


FIGURE 2.8: The Ontology for Run-Time Trustworthiness Maintenance (Gol Mohammadi et al. (2014)).

However, Ramanauskaite et al. (2013) uses the OntoMetric analysis method (Lozano-Tello and Gómez-Pérez (2004)) to assess how many of the concepts covered in 4 different security standards these two ontologies account for. These included ISO27001[12], PCI DSS (Payment Card Industry Data Security Standard), ISSA 5173 and NISTIR 7621[13]. They demonstrate that neither of these ontologies contain enough concepts or relationships to fully cover any of the analysed security standards. In response, they proposed an ontology which integrated concepts from Fenz and Ekelhart (2009) and Ramanauskaite et al. (2013), as well as the basic hierarchy of the organisation concept from the COBIT 5 framework[14], in order to achieve between 80-100% coverage of the security standards considered. The countermeasure classes were retained from Herzog et al. (2007).

In order to perform vulnerability analysis, Wang and Guo (2009) proposed the Ontology for Vulnerability Management (OVM) which captures the relationships between IT products, vulnerabilities, attackers, security metrics and countermeasures, among others. The model incorporates the concept of an active location (the location at which the

[12]https://www.iso.org/isoiec-27001-information-security.html
[13]https://csrc.nist.gov/publications/detail/nistir/7621/rev-1/final
[14]https://www.isaca.org/resources/cobit

software flaw, or vulnerability, manifests itself), alongside a richer model of the attack and attacker. They populate their knowledge base with data from the NVD (primarily using CVE entries) to demonstrate that an ontology for vulnerability analysis can be used to recommend appropriate countermeasures.

The 'Security Toolbox: Attacks & Countermeasures' (STAC) security ontology was proposed by Gyrard et al. (2013) and was intended to assist the development of secure applications. It represents the concept of a countermeasure in terms of its relationship to attacks, security properties and implementation method as well as its domain, in terms of the layer of the OSI model which it applies to. They also provide a web application which can interface with their reasoner to query the knowledge base.

Gao et al. (2013) use an ontology-based attack model to conduct security assessment from the attacker's perspective. They achieve this by categorising attacks using an attack taxonomy and integrate this taxonomy into a security ontology. This enables them to perform a security assessment based on measuring the effect of an attack on a system, rather than using traditional vulnerability analysis and a risk assessment.

Obrst et al. (2012) developed an ontology based on the the diamond model for malware analysis (Caltagirone et al. (2013)). Their ontology architecture spans a range of levels of abstraction corresponding to the ontology types introduced by Guarino (1997) and was developed principally through reuse of existing ontologies.

Oltramari et al. (2014) proposed three-layered ontology called 'CRATELO' comprising a top-level ontology 'DOLCE-SPRAY'[15]; a middle-level ontology known as SECCO[16] defining domain-specific concepts such as *threat*, *vulnerability*, *attack*, *countermeasure* and *asset*, and a low-level sub-ontology (OSCO[17]) which describes concepts related to cyber operations. Their implementation expresses countermeasures as a conditional rule formalised in CRATELO using SWRL which extends their OWL-DL axioms.

In order to aid cyber defenders to understand the security/cost trade-off between individual defences, Atighetchi et al. (2016) developed a reasoning framework called Attack Surface Reasoning (ASR) which comprises a set of ontologies for analysing the trade-offs with different compositions of cyber defences. The ASR framework uses ontologies to model attacks, defences, adversaries, systems and metrics, to identify attack vectors and compute the most appropriate cyber defences. This is achieved by incorporating the cost of defences into the defence model. They use Microsoft's STRIDE (Shostack (2014)) threat classification framework to classify attack steps, which are far less detailed than CAPEC and CPE.

[15] A simplified edition of DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering). It includes concepts such as *agent*, *object*, *action* and *task*.

[16] Security Core Ontology

[17] Ontologies of Secure Cyber Operations

The use of ontologies can be extended to incorporate the representation of security metrics, as proposed by Kotenko et al. (2013). Such metrics can be used to represent the cost and impact of countermeasures, metrics (such as the CVSS score of vulnerabilities), as well as the level of criticality of system components. Consequently, an ontological representation of security metrics can be integrated with other system and security components in order to develop a comprehensive decision support system which is suitable for generating a set of countermeasures to mitigate current threats.

Pendleton et al. (2016) suggested the Security Metric Ontology, as shown in Figure 2.9, which is used to measure system-level security metrics. It comprises four sub-ontologies (*vulnerability, attack, situations* and *defence mechanisms*) each of which corresponds to a set of metrics. Their study assesses the state-of-the-art to develop a taxonomy of system security metrics, which is then used to assess the limitations in existing security metrics. Since the ontology is primarily concerned with modelling metrics, it is less suited to performing reasoning and adopts some non-standard terminology. The OWL schema for their ontology has been published on GitHub[18].

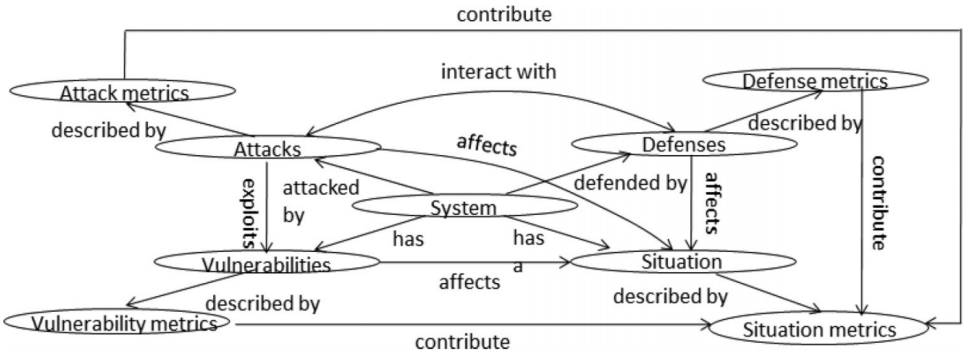


FIGURE 2.9: The Security Metrics Ontology (Pendleton et al. (2016)).

Syed et al. (2016) proposed the Unified Cybersecurity Ontology (UCO) which provides a loose high-level mapping between cyber security and other relevant ontologies. It is the first cyber security ontology to be mapped to general world ontologies, including DBPedia and the Linked Open Data cloud, as well as to aspects of STIX, CVE, CVSS, CAPEC and KillChain. This enables it to support a broad range of cyber security use cases. The UCO was authored by the original creators of STIX.

An important factor for assessing the effectiveness of a countermeasure is its coverage which relates to the proportion of the total attack surface which it positively impacts. This, together with the difficulty in trying to circumvent the countermeasure, as well as the number of attack phases which the countermeasure could potentially inhibit, enables the CyDef model to clearly characterise the effectiveness of cyber defences (Kimura et al. (2017)). However, the CyDef model provides no clear pathway to deriving such metrics, nor to representing them within an ontology.

[18]https://github.com/marcusp46/security-metrics-ontology

Qamar et al. (2017) proposed a threat analytics framework called the 'STIX-Analyser' which is an OWL-based ontology for gaining insights into cyber-attacks in order to assess their impact and associated risks. It uses STIX constructs, as well as a model of CVE entries, in order to assess the risk posed to a network from software vulnerabilities. The model does include a *CourseOfActions* class which uses the sub-classes *stage*, *efficacy*, *cost*, and *impact* to characterise a course of action, however the model is designed primarily for risk analysis rather than countermeasure effectiveness analysis, *per se*.

In addition to STIX constructs, although the VERIS framework is provided according to an original JSON representation, it can also be extended to be represented within an OWL-based ontology (Baesso Moreira et al. (2019)) which entails a range of benefits for incident handling and could also be used to facilitate reasoning using the VERIS enumerations.

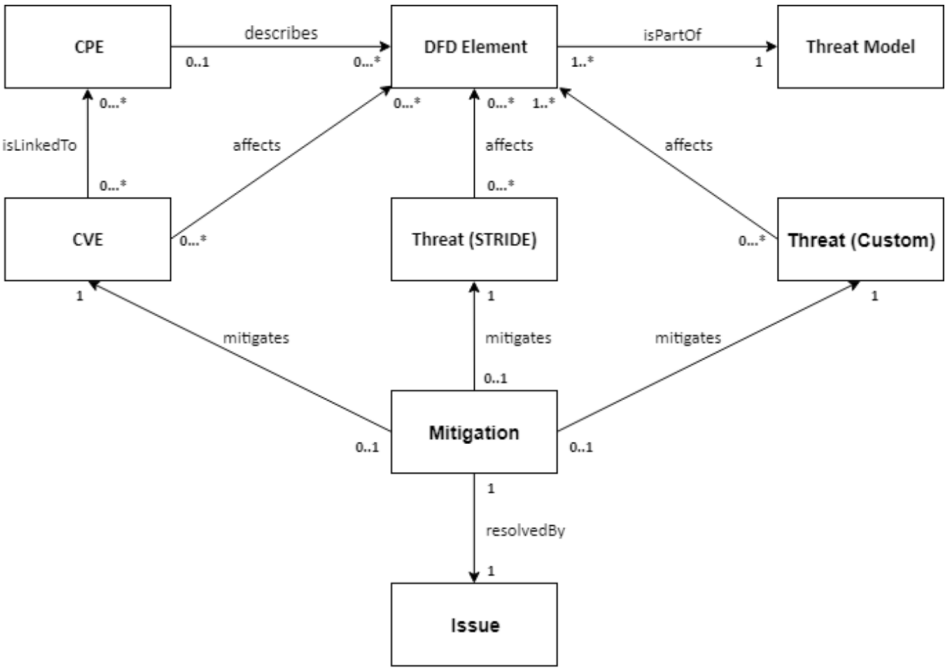


FIGURE 2.10: The OVVL conceptual data model (Schaad and Reski (2019)).

Schaad and Reski (2019) developed the 'Open Weakness and Vulnerability Modeller' (OVVL), illustrated in Figure 2.10, which comprises a conceptual data model based on the STRIDE threat types and DFD elements[19] in order to model the threats which emerge in software components during design. Its ontology includes a *mitigation* concept, however it is only used by the user to track threat mitigations while using the OVVL web-based tool. Their ontology uses the CPE database to describe DFD elements and maps them to vulnerabilities from the CVE database.

[19]Data Flow Diagram elements include 'process', 'data flow', 'data store' and 'interactor'

2.4.2 Threat modelling in the cloud environment

The use of virtual network topologies, such as those representing cloud-based services, can be helpful for the evaluation of security, including the selection of optimal countermeasures (Chung et al. (2013)). There are common features of cloud computing architectures which can be represented in a formal taxonomy Rimal et al. (2009) and various ontologies have also been proposed to unify cloud terminology in order to assist the selection of cloud service providers, as well as to model cloud services and their configuration (Nepal et al. (2012)).

Cloud services can be classified according to the service model they are based on, such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). They can also adopt various deployment models, such as public, private and hybrid cloud deployments. There are a range of threats which are particularly relevant to cloud-based services and it is possible to map these threats to the cloud service models to which they apply (Kalloniatis et al. (2014)). Further, Kalloniatis et al. (2014) also elicit a set of security and privacy properties which relate to cloud computing and show how they can be related to such threats. In addition, the way in which various security issues affect each service model and deployment model is well understood (Subashini and Kavitha (2011)).

A review of potential threats faced by cloud consumers was conducted by Hendre and Joshi (2015) which informed the identification of security controls which can effectively manage risks deriving from those threats. They developed a semantic model in the form of an OWL ontology describing threats, controls and compliance policies, which are used by a web-based recommendation tool for determining the kind of compliance necessary to respond to each cloud threat. Rather than assessing the impact of each control directly, their recommendation system is based on which set of compliance models each cloud service providers should adhere to. Further, they provide a very limited set of rules for reasoning using the ontology.

The use of a Security Reference Architecture (SRA) has been proposed as a way to guide the certification process of critical applications on the cloud to enable service providers to demonstrate trustworthiness and resilience against threats (Fernandez and Monge (2014)). Reference architectures (RAs) can be used to describe the main features of cloud systems, independently of implementation details and can therefore be used to inform the development of semantic models of the cloud environment, as exemplified in Figure 2.11.

Fernandez et al. (2016) provide a systematic methodology for producing SRAs for the cloud environment and show that they can be used to characterise threats and countermeasures in the form of misuse patterns and security patterns (Fernandez-Buglioni

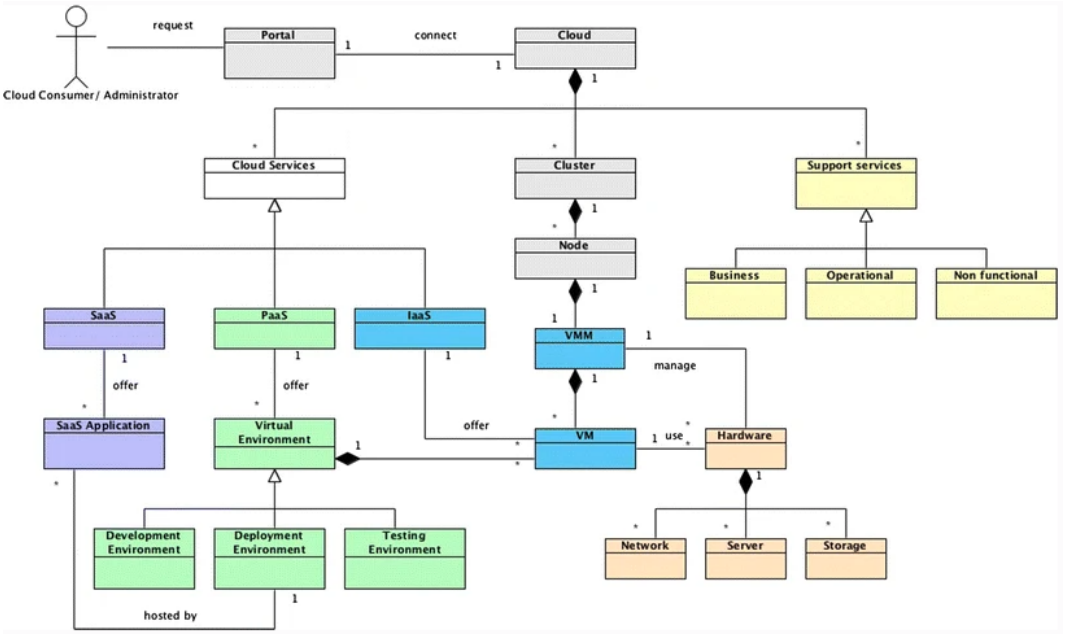


FIGURE 2.11: A class diagram of the cloud computing environment (Fernandez and Monge (2014)).

(2013)) respectively, as illustrated by conceptual diagram of Figure 2.12. Further, Taylor et al. (2020) have shown that recognition patterns based on asset configurations can also be used identify regulatory compliance, such as with GDPR, and to identify control strategies. It is also possible to use semantic modelling approaches to enable consumers to make effective security-based decisions about cloud providers without being familiar with their underlying technology (Hendre and Joshi (2015)).

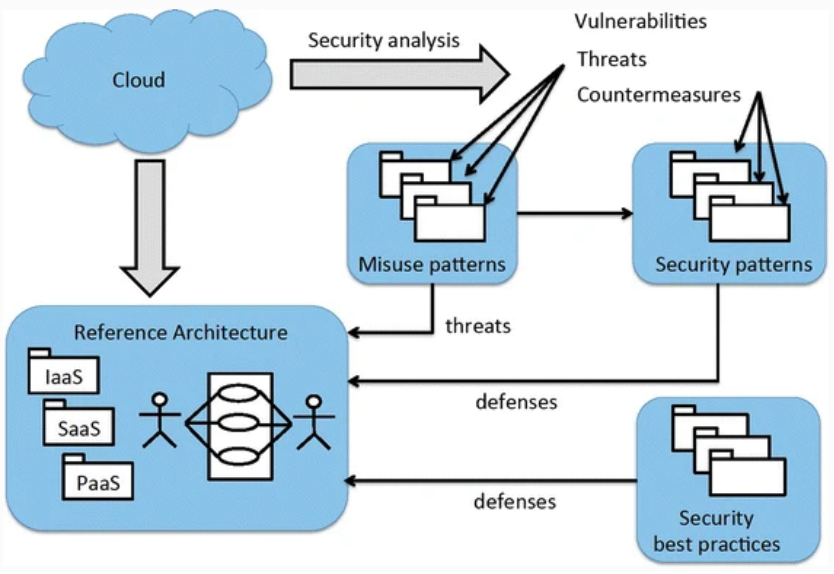


FIGURE 2.12: Conceptual diagram illustrating how misuse and security patterns can be used to analyse the security of a cloud service using a reference architecture (Fernandez and Monge (2014)).

Another security compliance ontology was developed by Zalazar et al. (2017) who focus on cloud service level agreements (SLAs). Their ontology specifies the terms, properties and relations of the cloud domain for evaluating their compliance with SLAs, and suggests further recommendations for cloud consumers when evaluating security risks. The security concepts are structured within the ontology in terms of how they influence the security properties of confidentiality, integrity, availability and accuracy. It is based on the security guarantees deriving from the compliance of the SLA with certain security regulations. They note that, since they are closely related to cloud computing, the ontology can be applied to fog computing by extending the ontology with additional concepts. Systems architectures involving the Internet-of-Things (IoT) also present threat modelling challenges as pointed out by Fadhel et al. (2019). They note that common approaches for threat modelling are insufficient on IoT environments due to differences in the kind of threats which IoT devices are subject to. These examples highlight the enduring relevance of threat modelling for contemporary network architectures, such as cloud-based systems.

2.5 Ontology Engineering

Ontology development methodologies generally adopt some combination of top-down analysis and bottom-up analysis when identifying concepts. A top-down approach to ontology development proceeds from the most abstract concepts to the most concrete (Lopez (1999)), which means that it requires understanding the semantics with which the end-users who will use the ontology will want to query the knowledge base (Obrst et al. (2012)). In contrast, bottom-up approaches are developed from the most concrete concepts up to the most abstract ones, requiring an understanding of the underlying data sources which are to be integrated with the ontology.

Uschold and Grüninger (1996) provide a compelling case for adopting a middle-out approach, in which one identifies the primary concepts with which the ontology is concerned first, allowing them to then specialise or generalise by using the aspects of top-down and bottom-up analysis as appropriate (Fernández-López et al. (1997)). Figure 2.13 shows the issues that are relevant in determining the general approach to use for ontology engineering.

Typically, a bottom-up approach involves a high level of detail which increases the overall effort. It can also reduce the ability to capture commonality, leading to inconsistencies and yet more rework being necessary. Although a top-down approach can reduce complexity, it can result in arbitrary high-level concepts, which reduces stability and also leads to more effort (Uschold and Grüninger (1996)).

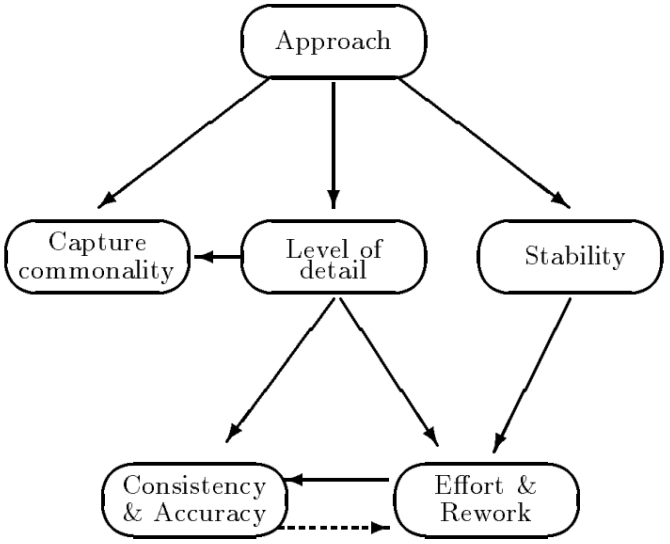


FIGURE 2.13: The issues which affect the ontology design approach selected, from Uschold and Grüninger (1996).

In contrast, Uschold and Grüninger (1996) suggests that a middle-out approach adopts an appropriate balance. Detail arises only as necessary by focussing on the most essential concepts, and higher-level concepts are defined in terms of these basic concepts, meaning that they are more likely to be stable.

When performing any of these types of analysis for ontology development, they are typically supported by the development of competency questions that represent the requirements of the ontology (Uschold and Grüninger (1996); Gruninger (1995); Gruninger and Lee (2002)). These are questions which the users of the ontology would expect the ontology to be able to answer once it is implemented. They are then used to identify the necessary concepts, properties, and relationships required for the ontology, and also used to evaluate the correct implementation of the ontology (Bravo et al. (2019)).

Obrst et al. (2012) remark that when defining an ontology, it can be useful to use existing schemas, dictionaries, and standards as a form of knowledge acquisition of the domain by identifying and analysing entities, relationships, properties, attributes, and other values.

2.5.1 Taxonomies and ontology development

A *taxonomy* is a hierarchical classification of entities within a domain, used to represent generalisation and specialisation relationships and subtype inheritance (Feilmayr and Wöß (2016)). In contrast, an *ontology* is broader than a taxonomy, in that it may represent more semantic meaning than just a classification scheme. Indeed, ontologies may comprise a full specification of a domain.

However, the foundation of an ontology still typically consists of a hierarchy of concepts as represented by a taxonomy (Guarino et al. (2009)) and an explicit taxonomy has been recognised as an important feature of ontologies (Pisanelli et al. (2003)). Thus, developing a domain taxonomy is an important step in the development of a domain or task ontology, and, in particular, in developing one that is capable of reasoning between layers of abstraction (Mavroeidis and Bromander (2017)).

Usman et al. (2017) propose an evidence-based taxonomy development method. Their approach is a revision of that proposed by Bayona-Oré et al. (2014) and is the result of a refinement process throughout the course of their own mapping study. The approach for evaluating taxonomies set out in Usman et al. (2017) is shown in Table 2.1. It represents a phased approach to planning, constructing, and validating taxonomies.

Phase	ID	Activity description
Planning	B1	Select and make clear the software engineering knowledge area for the new taxonomy
	B2	Clearly define the objectives and scope of the taxonomy
	B3	Describe the subject matter for the classification in detail
	B4	Select an appropriate classification structure type
	B5	Determine an appropriate classification procedure type
	B6	Identify appropriate data sources and data collection methods to provide knowledge related to the subject matter and taxonomy
Identification and extraction	B7	Extract the terms relevant to the new taxonomy from the collected data
	B8	Identify and remove redundancies and inconsistencies in terminology
Design and construction	B9	Identify and describe top-level dimensions as the main perspectives or top categories under which subject matter entities are classified
	B10	Identify and describe the categories for each of the dimensions
	B11	Identify and describe the relationship between dimensions
	B12	Provide guidelines for the adoption and evolution of the taxonomy
Validation	B13	Validate the new taxonomy through orthogonality demonstration, bench-marking and utility demonstration

TABLE 2.1: The taxonomy development methodology proposed by Usman et al. (2017).

Different classification structures have been used to construct taxonomies in various fields of knowledge. These structures include hierarchical, tree-based, paradigm-based or facet-based taxonomies (Kwasnik (1999)), however the main approaches to classification are considered to be enumerative and faceted (Rowley and Hartley (2000)). Enumerative taxonomies comprise fixed classes which makes classification intuitive, however it is often not possible to enumerate all classes in immature or evolving domains (Usman et al. (2017)). In contrast, faceted taxonomies specify the various aspects

of classes which can be extended or adjusted more flexibly. They tend to be based on the observation that classes can be viewed from many different perspectives (facets). Accordingly, facets (sometimes known as dimensions) may themselves have their own sub-classes.

The validation step is important for confirming the quality and utility of the taxonomy. It may be achieved in a number of ways, such as by demonstrating the orthogonality of the dimensions, by benchmarking against similar schemes, and by using the demonstration of utility, such as through a case study or expert opinion (Usman et al. (2017)).

2.5.2 Ontology design methodologies

It is common for ontologies to adopt a hierarchical taxonomic structure; however, an ontology is a much more flexible concept than a taxonomy and may assume a broader range of structures and complexity (Gruber (1995)). The way in which concepts within an ontology are structured is largely based on the purpose of the ontology model and the characteristics of the knowledge base that it must support (Hadzic et al. (2009)).

There is no standard methodology for building ontologies (Noy and Mcguinness (2001); Uschold and Grüninger (1996); Hadzic et al. (2009)), however, several methodologies have been proposed, each of which has slightly different strengths and intended use-cases (Corcho et al. (2003)).

Uschold and Grüninger (1996) published details of their Knowledge Engineering Methodology which offers a comprehensive methodology for developing ontologies. It adopts a middle-out approach and is domain independent; however, it contains very little detail as to its implementation (Lopez (1999)).

The METHONTOLOGY framework set out by Fernández-López et al. (1997) is a popular and detailed methodology, also adopting a middle-out approach, which contains both an ontology development process, as well as an ontology life cycle for evolving ontology prototypes. It uses a process involving specification, conceptualisation, formalisation and implementation to performing development activities, as well as various support activities, such as knowledge acquisition, evaluation and integration, in order to support the ontology development process. López et al. (1999) clarify that the framework is not intended to be implemented sequentially but that it should follow a degree of order to ensure consistency across each stage. In an assessment of various ontology engineering methodologies, Lopez (1999) find that Fernández-López et al. (1997) was the most mature methodology evaluated at the time.

The ontology proposed by Noy and Mcguinness (2001) has provided a lasting influence on other methodologies and is consistent with the general methodology of Fernández-López et al. (1997). Their ontology development methodology sets out the following steps to systematically engineer a useful ontology:

1. Determine the domain and scope of the ontology. Competency questions may be used here to define the purpose of the ontology, as well as to form the evaluation criteria once the ontology is complete.
2. Consider reusing existing ontologies. It may be beneficial to refine or extend existing models for the particular domain or task which this study is concerned with. In addition, ontology reuse is often a requirement, particularly if it needs to interact with other applications with a specific ontology or vocabulary.
3. Enumerate important terms in the ontology.
4. Define the classes and the class hierarchy. This is in essence the systematic construction of a taxonomy of the domain from existing knowledge (Feilmayr and Wöß (2016)). The hierarchy will often map on to the ontology type hierarchy of Guarino (1997)).
5. Define the properties of classes. Properties are sometimes referred to as slots or roles.
6. Define the facets of the properties. Facets are more commonly called restrictions. They are restrictions on the possible values of roles or properties which a class may have. Such restrictions may include the cardinality, value type or range of a property.
7. Create instances. By creating instances of classes in the ontology, one begins the process of building a knowledge base for the domain which can then be used to perform reasoning.

Noy and Mcguinness (2001) also provide three fundamental rules to ontology design. These are that a) there is no one correct way to model a domain, b) that ontology development is necessarily an iterative process, and c) that ontology concepts should be close to objects and relationships in the domain of interest, such as nouns (objects) or verbs (relationships) in sentences that describe the domain.

A longstanding discussion point on the definition of the ontology is related to whether axioms should be part of the ontology (Pisanelli et al. (2003)). The DOGMA[20] methodology (Spyns et al. (2008)) addresses this issue by separating specific concepts of an ontology from their axioms. An 'ontology base' layer contains the conceptualisation

[20]Developing Ontology-Guided Mediations of Agents

of the ontology domain, whereas an 'ontology commitments' layer contains the constraints, domain rules and mappings between elements of the conceptualisation. This enables users of the ontology domain to specialise the ontology base through modifications to the commitment layer, helping to render ontologies both highly reusable and useful, which are generally considered trade-offs.

The methodology described in Franz Baader et al. (2007) provides a helpful framework for developing ontologies through conceptual modelling based on Description Logics (DL) knowledge engineering. It begins by enumerating concepts which group values within the relevant domain, and developing a draft taxonomy of concepts. Although the ontology design process may involve revisiting this, it is clear that a suitable preliminary taxonomy of primative concepts is a precondition for the design of a conceptual model.

Obrst et al. (2012) provide a methodology for the reuse of existing ontologies, by (a) establishing ontologies in the domain areas of interest; (b) incorporating classes, properties and definitions from the best of the ontologies found in step a), and (c) if the number of classes and properties of an ontology from step a) grows large, considering directly importing the given ontology and establish equivalence relations between them and newly created ontology classes.

Bravo et al. (2019) provide an assessment of a range of ontology development methodologies, including METHONTOLOGY and the Knowledge Engineering methodology and determine that while METHONTOLOGY does provide a comprehensive and mature methodology, it can result in excessive complexity, particularly with regard to documentation, for large ontology systems. Instead, they present a simpler methodology for ontology engineering, as illustrated in Figure 2.14, which adopts an iterative approach to developing user-centred, modular and domain-oriented ontologies which uses competency-based evaluation.

Other methodologies exist, such as DILIGENT (Pinto et al. (2004)) for developing distributed ontologies in an iterative and collaborative manner; the SENSUS methodology (Swartout et al. (1997)) which maintains a common underlying ontology among all custom ontologies to facilitate knowledge sharing, and the TOVE methodology (Gruninger (1995)) which facilitates formal competency questions and verifying completeness.

In addition, Blanco et al. (2011) updates a previous systematic literature review (Blanco et al. (2008) to identify the key requirements that a unified security ontology should have. These requirements are identified as being static knowledge (natural language descriptions of essential concepts, relations and attributes), dynamic knowledge (the use of axioms to restrict the possible range of values, and facilitate deduction) and reusability (the selection of adequate concepts, which - as far as possible - exhaustively cover a domain). They note that providing this exhaustive coverage is difficult, but should be considered a priority when designing an ontology.

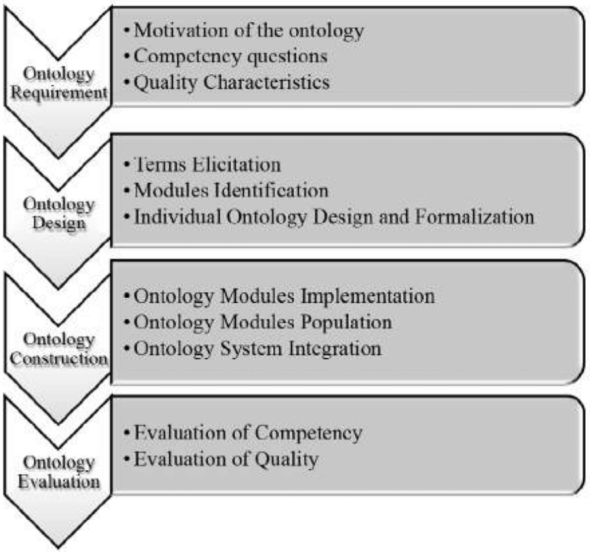


FIGURE 2.14: Phases and procedures of the ontology development methodology from Bravo et al. (2019).

2.5.3 Evaluating ontologies

The evaluation step set out by Fernández-López et al. (1997) provides a suitable addition to the activities in Noy and Mcguinness (2001). It defines an approach to iteratively verifying and validating the *evolving prototype* ontology. They suggest that this can be done by describing the techniques used to evaluate the ontology, the errors found in each activity and the sources of knowledge used.

Gruber (1995) identifies five design criteria for the development of knowledge sharing ontologies that represent an accepted design quality standard for threat ontologies. These criteria have proven to be useful as a guiding tool during the ontology design process (Hadzic et al. (2009)) and can also be used to validate ontologies, including in the form of expert surveys. The ontology should have the following qualities:

1. **Clarity**. It should effectively communicate the intended meaning of objective well-defined terms. Where a definition can be stated using logical axioms it should be. All definitions should be documented in natural language.

2. **Coherence**. The ontology should sanction inferences which are consistent with the definitions. Any informally defined terms should be consistent with formal concepts.

3. **Extendibility**. It should be defined with the use case in mind. In particular, it should be possible to define new terms using the existing vocabulary without needing to revise any existing definition. We suggest a slightly clarified term for this design criteria would be *extensibility*.

4. **Minimal encoding bias**. An *encoding bias* may arise when representation choices are made purely for convenience of notation. This can result in poorly defined and inextensible concepts. Therefore, concepts should be specified at the knowledge level, rather than the encoding level.

5. **Minimal ontological commitment**. Ontological commitment is based on consistent use of vocabulary and can be minimised by defining only those terms that are essential to the communication of knowledge consistent with the proposed theory. This enables it to avoid making unnecessary claims about the world being modelled, enabling the model to be easily instantiated as needed whilst maximising extensibility.

In addition to these five criteria, Gruber (1995) suggests that the extensibility and ontological commitment criteria relate to the notion of *representational adequacy* as referred to by McCarthy and Hayes (1987), however, there is little discussion on how adequacy can be attained. The information source(s) should have a suitable coverage of the breadth of threat information and the ontology must be sufficiently *expressive* in order to adequately represent the threat information once a knowledge base has been constructed. This can be validated with reference to the competency questions (Gruninger (1995)).

This concept of expressiveness is close to that of *richness* as set out by Tartir et al. (2005). They suggest a number of metrics for evaluating the quality of an ontology and of knowledge bases which use that ontology. These include the richness of the relationships, attributes and inheritance within the ontology. Although these metrics are not intended to represent an objective standard for the expressiveness of an ontology, they are helpful for evaluating certain aspects of its potential to adequately represent knowledge. Further, the *width* and *depth* of the structure of an ontology, as defined by Colomb (2002) are helpful for assessing expressiveness in terms of the average number of subclasses in each class, and the number of hierarchical levels in the ontology, respectively. In general, the complexity of a semantic model is often related to its degree of expressiveness (Franz Baader et al. (2007)).

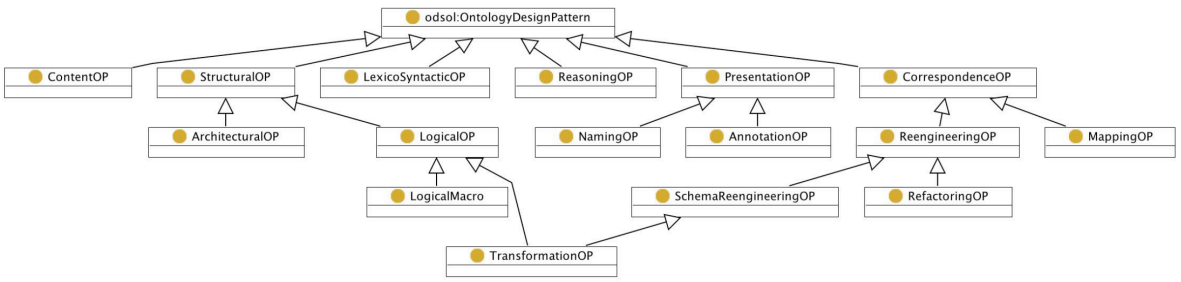


FIGURE 2.15: Types of Ontology Design Patterns from Gangemi and Presutti (2009), wherein the top layer represents the six OP families .

A set of Ontology Design Patterns (OP) were proposed by Gangemi and Presutti (2009) to address recurrent ontology design problems. These include a range of types of OPs, which are grouped into six distinct families, each with a particular ontology problem which they assist with. These OPs are shown in Figure 2.15 which are briefly described below.

1. *Structural OPs* address problems relating to the expressiveness and overall structure of the ontology.
2. *Reasoning OPs* assist in identifying the required reasoning to be performed on the ontology in order to execute queries or evaluation etc.
3. *Correspondence OPs* provide rules relating to transforming one ontology (a source ontology) into a new ontology, as well as with mappings between mappable elements.
4. *Presentation OPs* address the usability and readability of ontologies from the perspective of users.
5. *Lexico-Syntactic OPs* are linguistic structures which enable meaning to be derived using natural language.
6. *Content OPs* encode conceptual, rather than logical design patterns, to be used as basic building blocks within the domain.

2.5.4 The Semantic Web

The majority of data hosted on the internet is designed primarily for human consumption and is therefore substantially unstructured. The term 'Semantic Web' was coined by Berners-Lee et al. (2001) in which they set out a vision of what the internet could evolve into, characterised by establishing structure among meaningful content. The semantic web stack comprises a portfolio of technologies designed to utilise public information more effectively by facilitating machine reasoning. This challenge involves developing the ability to represent, acquire and utilise knowledge in semantic form (Sheth and Ramakrishnan (2003)).

The standards comprised within the Semantic Web are maintained by the World Wide Web Consortium (W3C) and are illustrated in Figure 2.16. A foundational principle of the Semantic Web is that it is made up of resources (or concepts) that are each identified by a Unique Resource Identifier (URI), of which URLs are a subset. The Semantic Web typically uses HTTP to transfer data and such data is represented using the Extensible Markup Language (XML) data format.

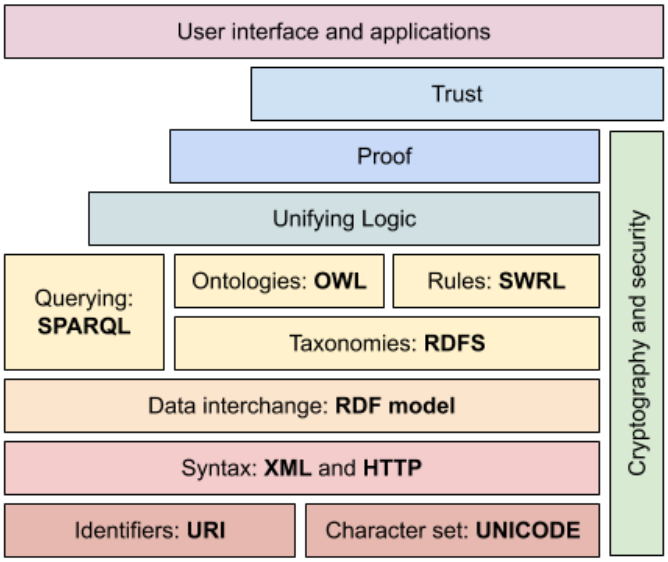


FIGURE 2.16: The technology stack of the semantic web.

The Semantic Web represents meaning in the form of semantic representation, namely, through use of words and the relationships between them. This is achieved using the Resource Description Framework (RDF) which is a model for representing meaning in the form of triples of the form *<subject, predicate, object>* whereby subjects and objects are concepts (which form nodes in an RDF graph) and predicates describe the relationships between the subject and object (and form the edges in the RDF graph). Each of the concepts has only one URI, enabling users to identify a concept and extract information relating to it. RDF knowledge graphs are queried using SPARQL, a recursive acronym for the 'SPARQL Protocol and RDF Query Language'. It can be used to represent questions to be asked of the knowledge graph in terms of the resources and predicates it contains.

The RDF Schema (RDFS) is a basic schema language to specify RDF content. Although many of the most widely used RDF vocabularies are specified in RDFS, it provides only a very limited way of expressing semantic meaning. The Web Ontology Language (OWL) however, is another widely used ontology language that provides additional benefits to RDFS, such as the ability to describe facts in terms of set operations, for example *<Example:Husband, owl:unionOf, (Example:Spouse, Example:Man)>*, as well as the ability to define constraints on how resources can be used. These additional features make it more appropriate for defining expressive ontologies with greater reasoning capabilities. OWL can be extended by using the Semantic Web Rule Language (SWRL) which enables rules to be specified which cannot be properly defined in OWL. SWRL can be used to specify more expressive relationships between OWL classes, such as for conducting cyber threat analysis (Surridge et al. (2013)). OWL comes in three distinct variants, each with different degrees of expressiveness: Lite, DL and Full. OWL-DL is

designed to maximise expressiveness while still retaining computational completeness, making ontologies specified in OWL-DL more decidable and efficient to compute.

There are a range of tools used to develop, test and utilise ontologies for the Semantic Web. The most widely-used open-source ontology editor and framework is called Protégé (Stanford University (1999)) which supports the W3C standards for the Semantic Web and fully supports OWL-DL as well as SWRL. Reasoning using OWL ontologies is performed by a reasoner which uses tableaux algorithms to create a tableau of the relevant facts and applies expansion rules to perform deduction and reasoning (Robinson and Voronkov (2001)). A commonly used reasoner is called Pellet which is an open-source reasoner written in Java that uses the Tableau Reasoner to perform reasoning on OWL-DL ontologies (Sirin et al. (2007)).

2.6 Conceptual model of research topics

The concepts which this review of the background literature is concerned with are contained within the conceptual framework of Figure 2.17. The framework is arranged to represent the activity of asset-based threat modelling for countermeasure evaluation. Blue boxes represent key components and resources of this activity; the red, orange and yellow boxes illustrate components of a threat model itself; pink boxes depict the main research questions which this research addresses; and the green boxes depict specific techniques used within our research methodology. The red lines and borders highlight the specific components of the conceptual framework to which our research contributes. For clarity, Figure 2.17 illustrates the mapping of the research questions described later in Chapter 3 to the corresponding components of the conceptual framework.

The conceptual framework represents a process for automated or partially-automated analysis of cyber threats and countermeasure effectiveness using asset-based threat modelling. It is a birds-eye representation of the topics which we discuss insofar as they relate to the research questions of this work. We include this illustration simply to assist the reader in understanding the structure of the literature, the significance of our research questions in their wider context, and the methodological techniques used to answer each question.

The conceptual framework involves four primary components which are typically considered in sequence: information sources; threat modelling; reasoning (including the generation of relevant metrics of relationships pertinent to threat modelling and analysis); and evaluation. Each of these components comprise a range of activities, methods and models which can be integrated together to provide a cohesive representation of a generalised asset-based threat modelling and evaluation process, particularly in the context of countermeasure analysis. The contributions made in this thesis apply to

a more general use-case, rather than specifically the task of countermeasure analysis, however, we have presented our work in this context to illustrate a primary use case which we are interested in.

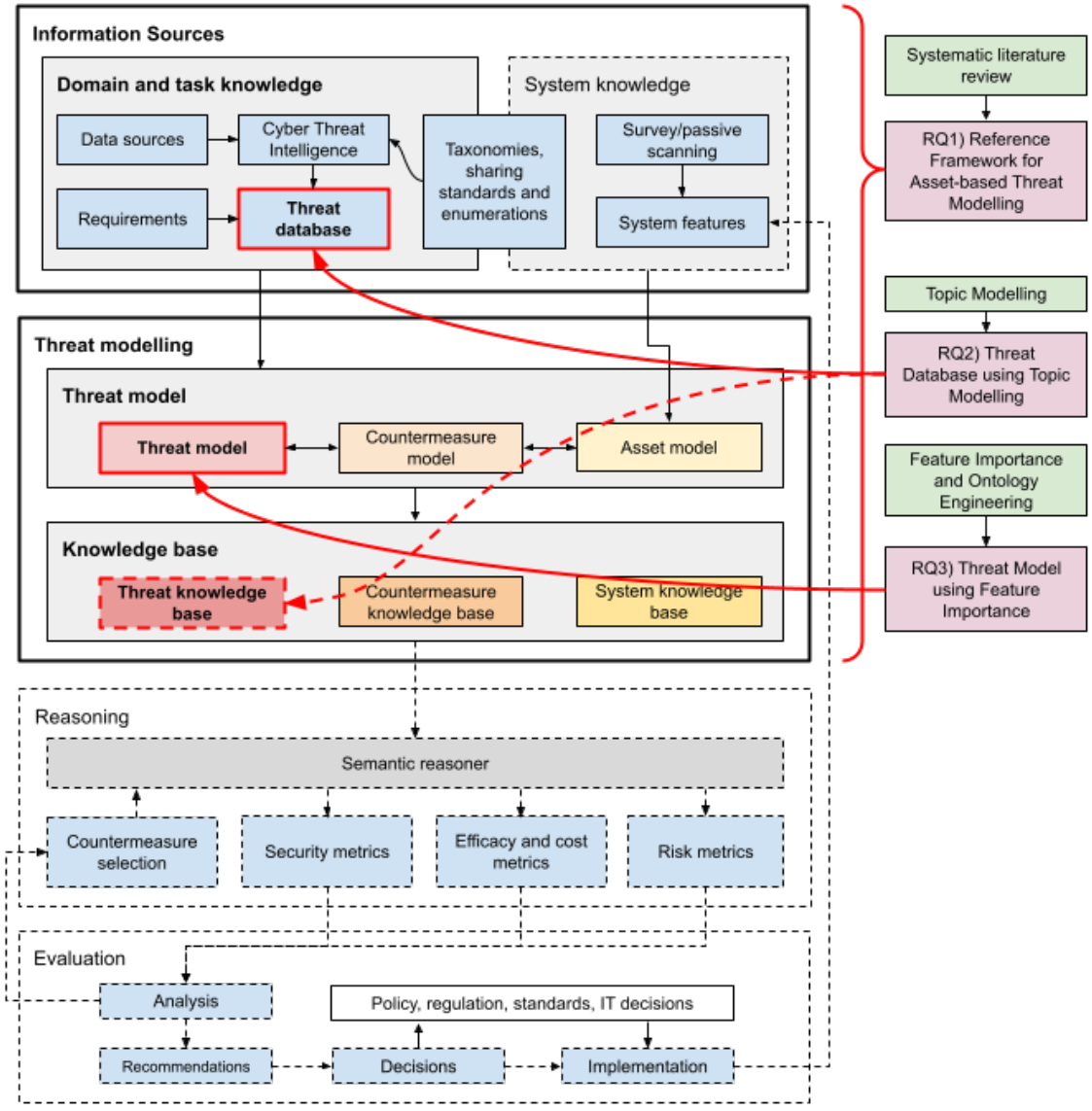


FIGURE 2.17: A conceptual framework illustrating the topics and methodologies which this review of the background literature discusses.

Figure 2.17 illustrates that asset-based threat modelling involves the characterisation of a target system in terms of a prior asset model. The *system features* refer to the features (such as network components and configuration) of the IT system under consideration, which may be elicited through use of a survey and which is consistent with the taxonomy underpinning the *asset model*.

Further, the countermeasures being evaluated in this use case must be identified and represented in a way that is consistent with the threat model and asset model. In other words, the selection of domain and task knowledge involves the selection of data (from

the *information sources*) on the basis of what is necessary to represent and assess the *security standard* (such as Cyber Essentials).

Each of these models may further comprise additional concepts which improve the reasoning capability of the semantic model. From the semantic model, a knowledge base may be instantiated by created instances of the semantic models, populated with data from the information sources. By translating the threat database into a structured representation, compatible with the threat model being used, one can arrive at a knowledge base of relevant cyber threat information. This knowledge based can then be used for higher-order analysis in the reasoning and evaluation stages.

The *reasoning* component comprises the collection of potential activities which involve reasoning using the knowledge base in order to derive answers to queries. These activities, such as calculating security metrics, assessing countermeasure efficacy and their impact on risk, are facilitated by the concepts contained within the semantic threat model and are typically represented in the form of an ontology query language. The *evaluation* stage represents the activities which may be performed automatically or manually in order to produce concrete recommendations regarding the most appropriate selection of countermeasures. Throughout each of these stages, due regard should be given to the relevant enumerations, taxonomies and sharing standards which together provide a coherent foundation to integrate these stages together.

2.7 Literature Critique and Research Gap

Having reviewed the relevant literature, the key areas which this study is concerned with are set out as follows, on the basis of the gaps in the literature.

1) Framework for asset-based threat modelling

The threat models described in Section 2.4, offer a range of proposals for using semantic threat modelling for performing countermeasure analysis. However, when evaluating the methodologies involved in developing those threat models, it is clear that there is no common framework to provide a baseline for the validity of such models, or to guide their development in a consistent manner. Indeed very few such papers mentioned any particular methodological rationale behind the design of their models whatsoever and most of the reviewed work simply focuses on isolated components of the threat model development process.

As such, the process of asset-based threat modelling (whether for performing countermeasure analysis or otherwise) is fraught with subjective design decisions and risks developing a threat model which does not fully capture all the essential elements of a valid asset-based threat model. It is certainly the case that this may simply reflect

the limited scope of previous work, however by developing threat models without the context provided by a reference framework for asset-based threat modelling, it is often ambiguous as to the limitations of a threat model which makes it hard to evaluate its completeness and validity. In short, there is no generalised reference framework to assist the integration, evaluation and continual development of a semantic countermeasure analysis process.

2) Techniques for producing a concise threat knowledge base

The sources of threat information which are used during asset-based threat modelling are also lacking in their coverage of the threat landscape, rendering the findings of such analysis to be inherently limited in its scope. In other words, for a cyber threat modelling process to be valid, it must be based on the analysis of a knowledge base of threat information which is suitably broad in its coverage to meet the requirements of the threat modelling activity. Without this, it is difficult to have much confidence that its findings are comprehensive and up-to-date. While there are existing open-source threat databases, such as the CVE, CWE and CAPEC enumerations, these are typically far too large to convert into a structured representation, such as an ontology's knowledge base, to analyse in a meaningful way using existing threat models.

In view of this, researchers tend to select a subset of threat entries from these databases to model, thereby reducing their coverage of the threat landscape. Even if the entirety of such a threat knowledge base is modelled, it is likely to quickly become out of date as new entries are added. Either way, there is a need to develop a technique for generating a consolidated threat knowledge base repeatably, without disproportionately compromising its coverage of the threat landscape.

3) Technique for quantitative cyber threat characterisation

There is also a lack of suitable techniques for characterising the attributes of a cyber threat model. While there are a range of existing techniques which can inform the development of threat models, such as the ontology engineering methodologies discussed in the next chapter, there are no known quantitative techniques for identifying the particular threat attributes (i.e. ontology concepts) which are pertinent to the creation of a threat model. Hence, threat models are typically - at least to some degree - the subjective expression of the experience and intuition of the designer. This means that the academic justification for the selection of threat attributes to include in a given threat model using existing methodologies is comparatively weak. Therefore, there is a need to develop a new technique for automatically generating robust quantitative data for evaluating the relative importance of the attributes of a cyber threat in order to perform characterisation.

2.8 Summary

This chapter set out a conceptual model of the domains relevant to this study. It has reviewed the cyber risks posed to SMEs and the techniques used to model the IT network of SMEs. This was followed by a review of various aspects of CTI, asset-based threat modelling and associated ontologies. As alluded to, this work utilises topic modelling and feature importance. These techniques, along with a review of their use in cyber security, are set out in Chapter 3.

We have also explored the literature on taxonomy development, ontology engineering and semantic web technologies. The following observations have been identified to help answer the research questions of this study:

- The taxonomy development methodology proposed by Usman et al. (2017) clearly describes an approach suitable for developing a taxonomy of cyber threats or cyber threat attributes. However, it does not offer sufficient support for identifying candidate categories.
- A cyber threat taxonomy is beneficial for classifying threats, but is also an important milestone in the development of a cyber ontology for countermeasure analysis since they are typically synergistic with the early stages of many ontology development methodologies.
- The ontology development approach proposed by Noy and Mcguinness (2001), and the evaluation activities set out by Fernández-López et al. (1997) and Gruber (1995) provide a suitable foundation for an iterative methodology to develop a cyber ontology for countermeasure analysis. These methodologies are used to inform the ontology engineering process defined in Chapter 6.
- Developing a robust ontology for modelling a knowledge domain is difficult, and no such model will exhaustively characterise its intended domain. Every model entails constraints and limitations, and those limitations should be determined based upon the use-case of the model.

Having conducted a review into these areas, we have identified that the background literature fails to:

- Present a robust framework for guiding and evaluating the process of asset-based threat modelling.
- Offer a repeatable technique for generating a concise, expressive and broad threat knowledge base.

- Describe a robust technique for evaluating the importance of threat attributes for characterising a threat model.

This study seeks to rectify these deficiencies in the literature by contributing a reference framework for asset-based threat modelling, as well as techniques for threat knowledge base generation and quantitative threat characterisation.

The primary ambition of this study is therefore to enable the field of threat modelling to continue to develop in order to facilitate, among other things, the effective evaluation of cyber controls (such as those set out in the Cyber Essentials scheme) and to motivate greater interest in this field for future work.

Chapter 3

Research Methodology

In view of the gaps in the relevant literature identified in Section 2.7, this chapter defines the research questions which this thesis aims to answer and discusses them. We also identify and justify the methods and techniques used to address these research questions, and explain how they have been used and validated during this research.

3.1 Research Questions

There are a number of studies that have sought to model cyber threats for a broad range of use cases. However, our literature review has identified that there is currently no robust and effective framework for enabling researchers or professionals to develop asset-based threat models, particularly for the purpose of evaluating the effectiveness of cyber countermeasures.

Therefore, the central concern of this research project is to improve the quality and capability of asset-based threat models to address these questions. In view of this, several research questions have been established that together amount to a suitable response to the research gaps described in Section 2.7.

In order to address the first research gap, our first research question is defined as follows:

RQ1: What activities are necessary to sufficiently characterise a robust and generic asset-based threat modelling process?

Keyword analysis of publications relating cyber threat modelling has been used to identify the main clusters of activity in any threat modelling process. An analysis process based on a systematic literature review is also used to validate these main activities and to validate a generic asset-based threat modelling process. The systematic

literature review methodology as set out in Kitchenham and Charters (2007) and as summarised in Silva and Neiva (2016) is used to implement this analysis. The results from this Chapter constitute a Reference Framework for Asset-based Threat Modelling (ReFAThM).

In addition to addressing RQ1, we are also interested in identifying ways in which existing data sets, such as the CVE and CWE data sets, can be used to generate a concise threat taxonomy in order to address the gaps identified in Section 2.7. In view of the advances made in the field of natural language processing (NLP) and transformer-based large language models (LLMs) in recent years, and their applications in the field of threat modelling (Elsharef et al. (2024)) we identified NLP as a suitable field for addressing this challenge. Thus, we formalised our second research question as follows:

RQ2: How can existing techniques in natural language processing (NLP) be used to derive a consolidated cyber threat knowledge base from an unstructured threat database?

The primary activity in conducting asset-based threat modelling is the design and evaluation of the threat model itself. This process involves making a range of design decisions about what concepts constitute a cyber threat. However, existing research does not provide a suitably robust method for identifying the attributes which constitute an appropriate characterisation of a cyber threat. Therefore, our third research question is:

RQ3: To what extent can feature importance in classification be used as a quantitative measure of the relative importance of cyber threat attributes for the development of a cyber threat model?

The results of RQ3 are a quantitative evaluation of the relative importance of each of a number of candidate threat attributes which can be used to justify the selection of threat attributes to include within a given asset-based cyber threat model.

3.2 Relevant Research Methods

Due to the range of research activities which this project involves, a variety of different methods are considered appropriate to address the research questions. In addition to the research methods discussed in this Section, Chapter 2 provides a detailed discussion of the relevant literature concerning taxonomy development and ontology engineering which are important parts of the methodology of this study.

Most research methods can be broadly characterised as being either *quantitative* or *qualitative* methods, or a mixture of the two.

3.2.1 Quantitative methods

Quantitative research methods use numerical data in order to understand a particular phenomenon (Gay (1996)). This typically involves limiting the effects of human bias in extracting data in order to describe, predict or to control that phenomena (Taylor (2005). The evaluation of such data is often conducted using descriptive statistics which involves determining the characteristics of a specific phenomena. Alternatively, inference techniques can be used to determine whether characteristics of specific examples of certain phenomena can be generalised to provide more widely-applicable insights (Taylor (2005)). In computing and information security a range of quantitative methods have been used for measuring and evaluating concepts (Basili et al. (1994).

3.2.2 Qualitative methods

In distinction to quantitative methods, qualitative research methods seek to understand phenomena through use of descriptive techniques (Taylor (2005)) by answering questions on ”what” those phenomena are and ”how” they behave (Yin (2017)). They should generally be analysed using inductive or deductive thematic analysis (Braun and Clarke (2006)), such as using expert reviews, interviews or focus groups.

3.2.3 Mixed methods

It is common for the benefits of both quantitative and qualitative techniques to be combined by being used together in order to collect and analyse data relating to a single research question (Lister (2005)). This can enable statistical and descriptive techniques to be used to inform and validate each other (Creswell and Creswell J. D (2017)).

An example of a mixed methods technique is a **case study** which uses a particular interesting example in order to gain understanding or test a particular hypothesis (Stake (1995)). Although they can be perceived to lack rigour, they can be extremely helpful depending on the nature of the question, and are also very flexible (they can be explanatory, exploratory or descriptive) (Yin (2017)). Yin (2017) also presents five design features of a high quality case study which can be used to ensure rigour, including a study question, the propositions, relevant units of analysis, the logic linking the data to the propositions, and the criteria for interpreting the results.

Another example of a mixed method approach might be a systematic literature review. Kitchenham and Charters (2007) identify that a rigorous systematic literature review process should involve steps such as defining the main research question, relevant keywords and a search string; identifying suitable scientific databases to search; refining

and executing the string; defining and applying inclusion and/or exclusion criteria and extracting the relevant answers from each selected paper.

As discussed in Xiong and Lagerström (2019), the most common forms of validation of threat models or threat modelling methods are through theoretical examples, such as simulations or analysis, or through empirical case studies, including implementations and expert reviews.

3.3 Topic Modelling with Text-based Datasets

Topic modelling is an NLP technique used to automate the process of identifying latent topics in a text corpus and is pertinent to our approach to answering RQ2. The goal of topic modelling is to uncover the hidden thematic structure of semantic concepts in a large collection of documents, typically in order to organise, understand and summarise large text-based datasets (Zhao et al. (2021)). Whereas traditional clustering is designed to identify groups (i.e. 'clusters') of data points within a quantitative dataset, topic modelling is suitable for clustering text-based data.

Traditional models, such as Latent Dirichlet Allocation (LDA) tackle this challenge using Bayesian probability with a *bag of words* approach (Blei et al. (2003)). While LDA-based techniques have become commonplace in the field of topic modelling, they do have some notable limitations. In particular, they do not utilise embeddings to gain semantic insight in the relationship between words; they struggle to understand the context and implications of word ordering in a text (Rashad et al. (2022)); and they struggle to accurately represent the meaning of shorter texts (Egger and Yu (2022)).

The Top2Vec topic modelling algorithm has gained prominence due to its advantages over LDA (Angelov (2020)). Top2Vec utilises the techniques in Word2Vec and Doc2Vec to transform words and documents into a shared high-dimensional feature space using embeddings (Le and Mikolov (2014)). The advantage of this is that the topic model can represent texts as fixed-length embeddings, even if the original texts have different lengths. Top2Vec is able to capture the depth and nuances of sentences and identify similarities in meaning between words and phrases. The primary limitation of Top2Vec, however, is that it requires dimensionality reduction to be performed using Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. (2018)) and that clustering be done using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) (McInnes et al. (2017)).

More recently, BERTopic has taken the additional step of incorporating the BERT (Bidirectional Encoder Representations from Transformers) model to generate enhanced processing of natural language. In particular, BERT performs better at tasks which require an even more nuanced understanding of semantic meaning of word ordering

and context. Devlin et al. (2018) describe that it is also able to more accurately distinguish between different meanings of words based on their usage, and is also proficient even when dealing with short documents. The BERTopic technique consists of the four main steps:

1. Embeddings: Each document is converted into a numerical vector using pre-trained BERT models. These embeddings capture the unique fingerprint of each piece of text.
2. Dimensionality reduction: Embeddings are high dimensional vectors which are computationally taxing to process. Therefore it is advantageous to reduce the number of dimensions to allow for more efficient processing and classification. The goal is to minimise the dimensionality of the document embeddings whilst retaining a representation of its essential features.
3. Clustering: This stage groups similar topics together based on their embeddings. This step categorises the data into semantically similar clusters, allowing for easier analysis and interpretation.
4. Topic representation: Having identified the main clusters, the latent semantic concepts represented by the centroids of those clusters are identified by converting representative embeddings back into a text-based value. This enables the topics within the data to be summarised in a human-readable manner for further analysis and evaluation.

While the default settings utilise UMAP and HDBSCAN as with Top2Vec, BERTopic allows the user to select from a variety of compatible dimensionality reduction and clustering algorithms, such as Principle Component Analysis (PCA) (Jolliffe (2002)) and K-Means. It also offers a novel technique called c-TF-IDF (class-Term Frequency-Inverse Document Frequency) for topic representation which improves upon previous techniques. We assess that BERTopic provides a comprehensive set of tooling for categorising and quantifying documents, including those relating to cyber threats. When evaluated using a variety of primary datasets, the current consensus is that BERTopic offers superior performance to Top2Vec or earlier topic modelling algorithms at identifying more coherent and distinct topics (Egger and Yu (2022)).

3.3.1 Topic modelling in Cyber Security

There are a few instances in the existing literature of topic modelling being used in the field of cyber security in a variety of contexts. In particular, Kolini (2017) uses clustering and topic modelling methods to evaluate the differences between 60 national cyber security strategies. They identify a range of valuable insights into the deficiencies

of such strategies, which appear to validate the utility of using topic modelling for understanding and analysing text-based documents.

In a separate study, Hossen et al. (2021) also collected data from a hacker forum (nulled.io), and used LDA and Non-negative Matrix Factorisation (NMF) to conduct topic modelling. This work was later extended by Suryotrisongko et al. (2022) who implemented topic modelling approaches for the same task using both BERTopic and Top2Vec to extract open-source intelligence (OSINT) from the same hacker forum, this time as part of the Open Web Application Security Project (OWASP) Maryam OSINT framework. However, neither of these projects use topic modelling to characterise the entire domain threat information and don't conduct feature importance analysis to characterise cyber threats.

Sleeman et al. (2021) apply topic modelling to time-stamped cyber security document collection to show how the significance and details of concepts found in them evolves over time. These concepts are represented in a semantic knowledge graph to support interrogation, inference and discovery. They demonstrate that it is possible to use topics as a bridge to relate documents across corpora over time. While this study addresses a different set of challenges than ours, this is an important project to demonstrate the feasibility of using topic modelling to construct a structured knowledge graph of threat information to facilitate semantic reasoning.

3.4 Feature Importance Analysis

In order to address RQ3, we use feature importance analysis in order to quantify the importance or degree of contribution of each identified attribute of cyber threats. This analysis aims to determine the relevance and predictive power of each feature in categorising cyber threats based on their text-based descriptions. Whereas the technique of feature importance is a well established task in supervised learning, such as classification, it is less well explored in the context of unsupervised learning, such as clustering.

In our review of the literature on feature importance for clustering, we found that they were either focused on the pre-clustering stage as, in the case of Goswami et al. (2017), rendering them irrelevant to our goals, or were model-specific techniques, as with Cabezas et al. (2021). We identified some model-agnostic approaches which used permutation techniques to measure feature importance (Scholbeck et al. (2022) and Ellis et al. (2021)). This involves varying the values of each feature of a given document and measuring the corresponding different in clustering accuracy, thereby being able to determine which features are more critical for accurately classifying a given document. However, these techniques appear to require considerable computational resources and there is an insufficient coverage of this approach in the literature to make it feasible.

Ismaili et al. (2014) presents an alternative technique for evaluating the contributions of explanatory variables to a clustering. This is achieved by training a classifier to predict the instance membership to the clusters using each individual variable. The predictive power of each variable (feature) is then measured using two evaluation criteria and the variables are sorted accordingly. This enables feature importance to be deduced from these importance metrics used in the classifier. The work of Badih et al. (2019) discusses the use of variable importance metrics such as the mean decrease impurity in Random Forest and XGBoost, as well as weight coefficients in Support Vector Machines (SVMs). Combining these techniques to conduct feature importance is judged to be both feasible and methodologically robust, repeatable and generalisable to other domains. Therefore, this classifier-based approach is determined to be the most suitable to answer RQ3.

3.5 Research Methodology

The selected research methods as mentioned in Section 3.1, along with their associated research activities, are illustrated in Figure 3.1. This methodology comprises 5 major areas, corresponding to the primary domains of the conceptual framework of Figure 2.17.

The first research activity comprises a literature review of the research topics pertinent to our research questions and a discussion of the research gap. This process results in a conceptual framework which is illustrated in Figure 2.17 and the key findings discussed in Chapter 2.

The second research activity addresses RQ1 by conducting a keyword analysis in order to identify the key activities of a threat modelling process. These are then improved and validated using a systematic literature review process resulting in the development of a Reference Framework for Asset-based Threat Modelling (ReFAThM) as discussed in Chapter 4.

In the third research activity, we use natural language processing to answer RQ2. As discussed in Chapter 5, this involves acquiring CWE entries as our primary data source and and pre-processing them using GPT-3.5. We then use topic modelling followed by a cluster merging step in order to identify 19 threat types which represent the full breadth of the threat landscape covered by the initial CWE entries. These threat types are useful for a broad range of purposes in research and industry and can be re-generated quickly on a routine basis as threats in the primary data sources evolve over time.

In the fourth research activity, the threat types (i.e. clusters) identified in the previous activity are used to conduct features importance analysis. This process results in a quantitative evaluation of the relative importance of each of the threat attributes comprised within the topic modelling stage and enables us to answer RQ3. Accordingly,

we are then able to propose an ontology as a 'core' threat model on the basis that the most important threat attributes are the attributes necessary to properly characterise a cyber threat. This activity and its results are discussed in Chapter 6.

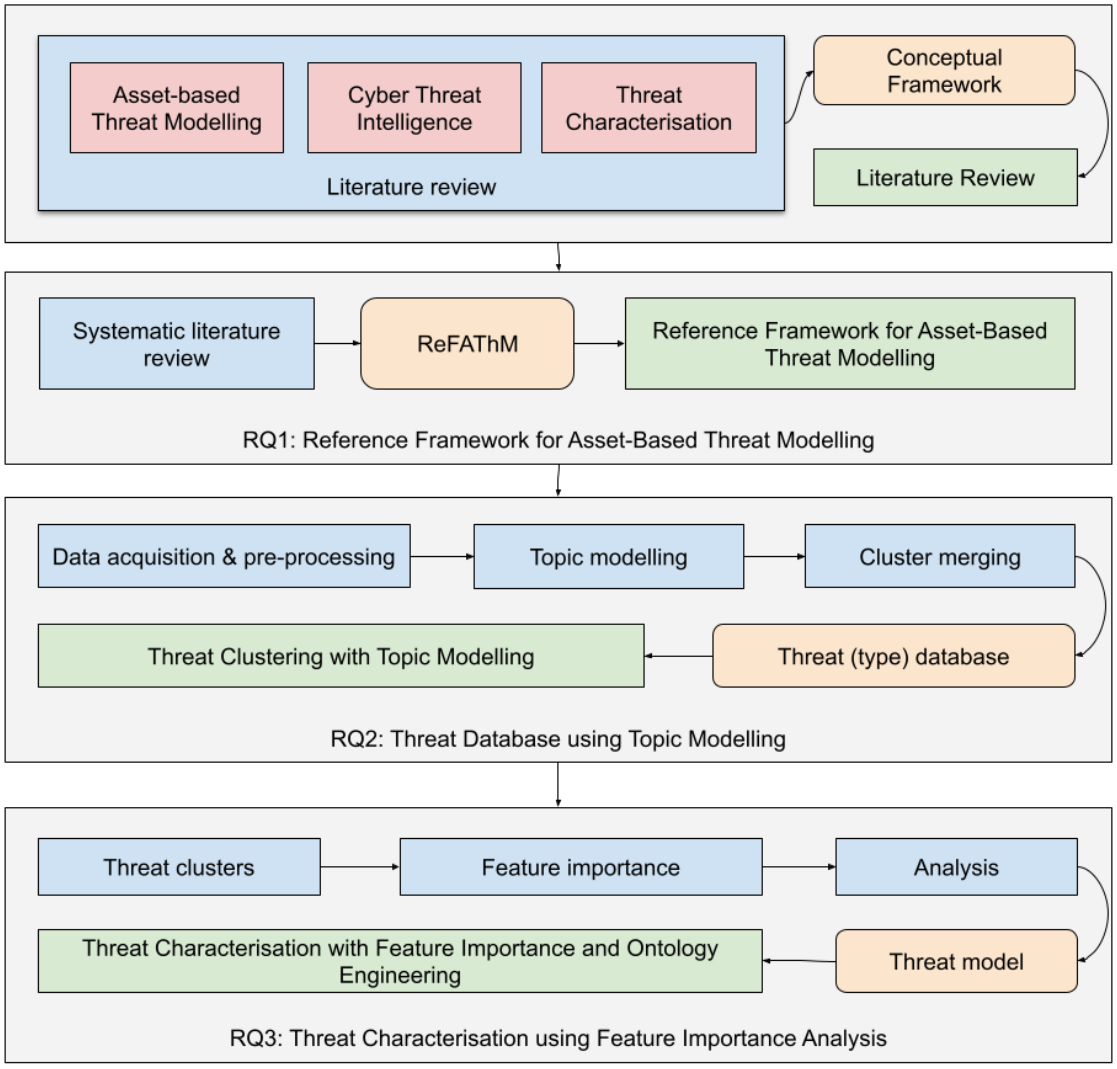


FIGURE 3.1: Research activities and methods. Research activities are highlighted in blue, research outputs are highlighted in yellow and the corresponding chapters in this thesis are highlighted in green.

Chapter 4

A Reference Framework for Asset-based Threat Modelling

This chapter presents an asset-based threat modelling framework. It is derived from an analysis of threat modelling publications and validated through a process involving a systematic literature review (SLR). The threat modelling framework is therefore generic, in that it relates to a broad range of potential embodiments of asset-based threat models, but it also has a structure which provides a degree of direction over how a threat model should be developed.

The process used to develop our threat modelling framework begins by analysing the keywords associated with academic publications in the field of cyber threat modelling and clustering them into respective threat modelling components. This step is used to identify the key aspects of a generic threat modelling process and to produce an initial enumeration of threat modelling activities.

This draft threat modelling framework is then evaluated using an SLR process to determine the suitability and coverage of each activity in constituting a generic asset-based threat modelling process. In response to the findings of the SLR process, a number of amendments are made to the threat modelling framework, and the amended framework is evaluated again using the SLR process.

A discussion is provided which summarises the main findings of this analysis process and the asset-based threat modelling framework is visualised to facilitate its comprehension and application. This chapter addresses research question 1 as defined in Section 3.1, namely to determine what activities are necessary to sufficiently characterise a robust and generic asset-based threat modelling process.

4.1 Identifying key cyber threat modelling activities

In order to minimise the number of assumptions which need to be made in order to characterise a generic asset-based threat modelling framework, we used a process for analysing the keywords in academic publications, and their relationships to each other, in order to identify clusters of keywords representing the domain of cyber threat modelling. This process involves using the OpenAlex dataset (originally based on the now redundant Microsoft Academic Graph database) to extract keywords from the abstracts of relevant scientific papers. These keywords are stored in an ElasticSearch index which can be queried. The search query '`cyber AND threat AND model`' is used to extract relevant articles and generate a '.wos' file from its results. A list of relevant keywords are then identified for each of those articles, and a clustering process is used to identify N subgraphs. Associated map and network files are generated to enable VOSviewer to be used to visualise those graphs of clustered keywords. The visualisation output of VOSViewer of the keyword analysis is illustrated in Figure 4.1.

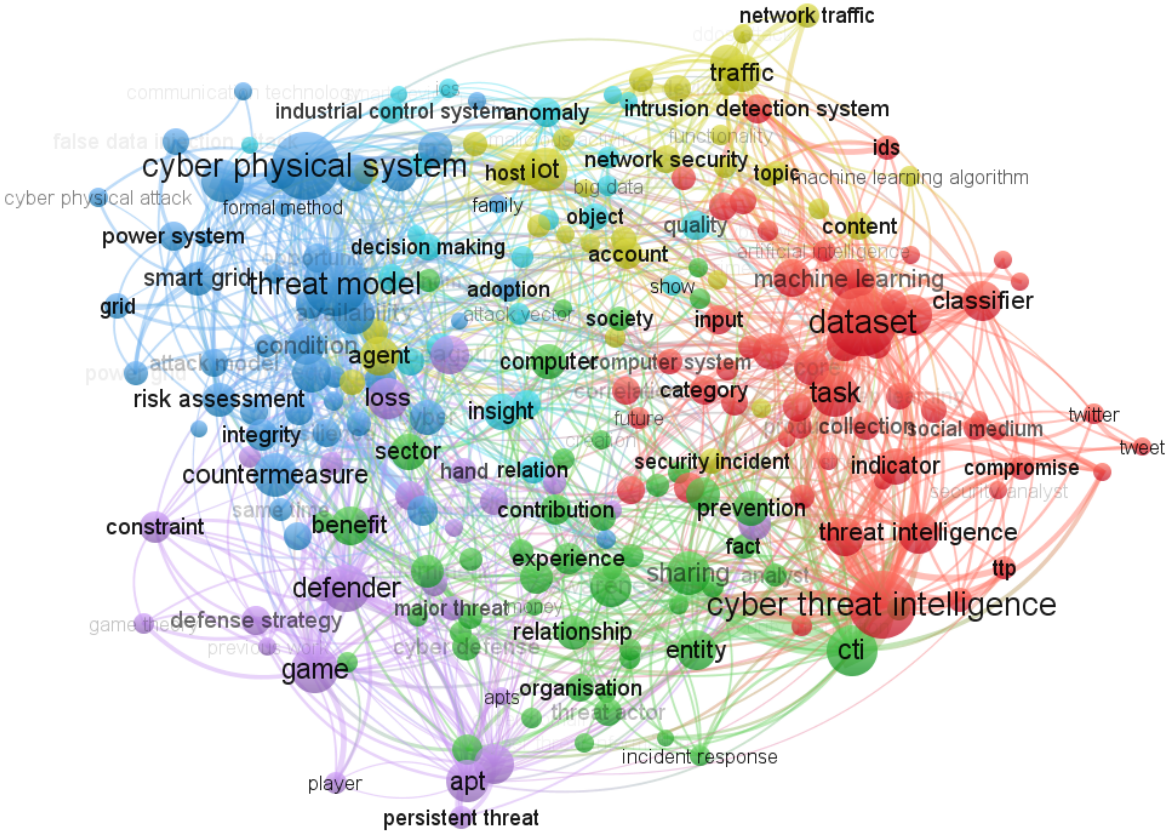


FIGURE 4.1: An illustration of the the keyword analysis and clustering performed on papers relating to threat modelling in the Microsoft Academic Graph publications database, as visualised using VOSViewer. A more detailed version of this graph is illustrated in Figure A.1 of Appendix A

.

The extracted keywords can be classified into four distinct classes, each of which corresponds approximately to one or two of the coloured classifications identified by VOSviewer,

as shown in Figure 4.1. These classes are *modelling* (red and green clusters combined), *IT assets* (yellow cluster and part of the blue cluster), *threats* (blue cluster) and *controls* (purple cluster). While the classifications identified by VOSviewer do not correspond precisely to these four classes, each of the keywords can be appropriately classified into one of them. Consequently, these four classes can be understood to represent an initial high-level characterisation of the threat modelling domain.

As shown in Figure 4.2, the four classifications of threat modelling keywords are used as the basis for the initial enumeration of threat modelling activities. In particular, the *threats* and *controls* classes have been separated in the same way into an identification step and a characterisation step. These initial activities are further supplemented by activities for *defining the goal* and *validating* the threat model, which are considered inherent activities in any rigorous threat modelling process (Meszaros and Buchalcevova (2017)).

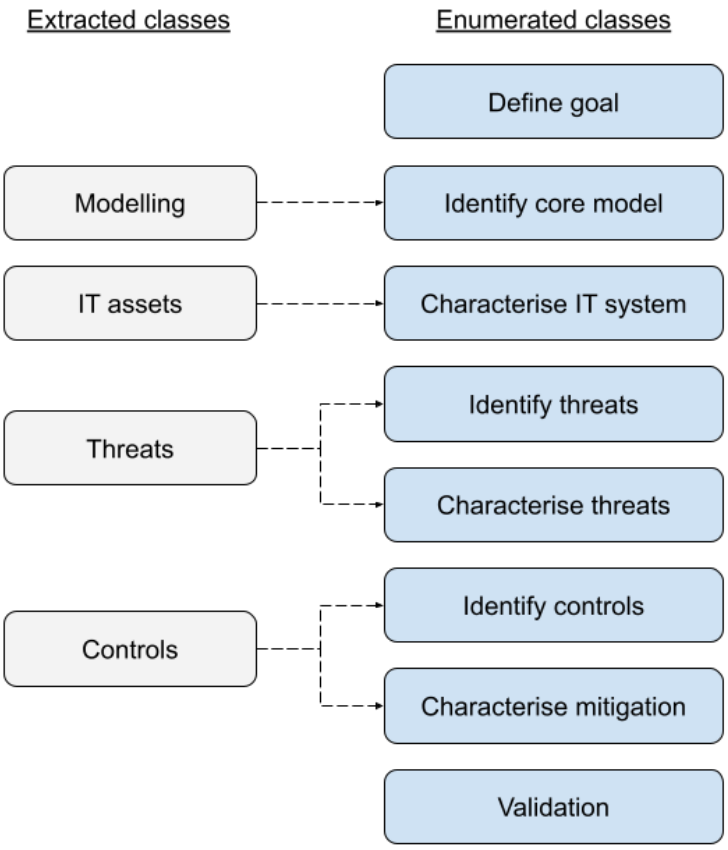


FIGURE 4.2: An illustration of four classes extracted from the keyword analysis process and their mapping to the draft threat modelling activities.

The requirements of each of the activities of the draft threat modelling framework, for the purposes of the SLR, are defined as follows:

1. **Define goal:** the purpose of the threat model, or threat modelling method or process, should be clearly defined.

2. **Identify core model:** the essential concepts necessary to characterise a threat must be identified, and the relationships between them established.

3. **Characterise IT system:** the essential nature of the IT system must be expressed, such as, in terms of the flow of data, the IT assets involved, or the properties of the IT environment within which threats may emerge.

4. **Identify threats:** the name or other identifier of at least one threat which is relevant to the IT system must be provided.

5. **Characterise threats:** the essential nature of the threat must be expressed with reasonable detail and clarity, in terms of how it is manifested within the IT system.

6. **Identify controls:** the name or other identifier of at least one mitigation to one of the aforementioned identified threats must be provided.

7. **Characterise mitigation:** the essential nature of the aforementioned control(s) must be expressed with reasonable detail and clarity, in terms of the effect it has in mitigating a corresponding threat.

8. **Validation:** the rationale underpinning the implementation of the threat model, or threat modelling process, should be validated using a suitable technique, such as using a case study, expert review, semantic modelling, ontology engineering or using formal methods.

4.2 Evaluating the draft framework

The analysis of the draft threat modelling framework utilises a process based on a systematic literature review in order to evaluate the relevance and sufficiency of the proposed threat modelling activities to the generic threat modelling process. The systematic literature review was performed in accordance with the methodology set out in Kitchenham and Charters (2007). The process of our particular systematic literature review is shown in 4.3.

The systematic literature review portion of this analysis task was executed using four scientific databases: IEEE Xplore, Scopus, SpringerLink and Web of Science. While many other databases are available for inclusion, we deemed these to offer a suitable coverage of the pool of relevant literature and so others were excluded in order to limit overlap, optimise for database relevance and to constrain the scope of the SLR process. The searches were executed on 14 March 2022 using the search string:

```
"threat modeling" OR "threat model" OR "threat modelling"
```

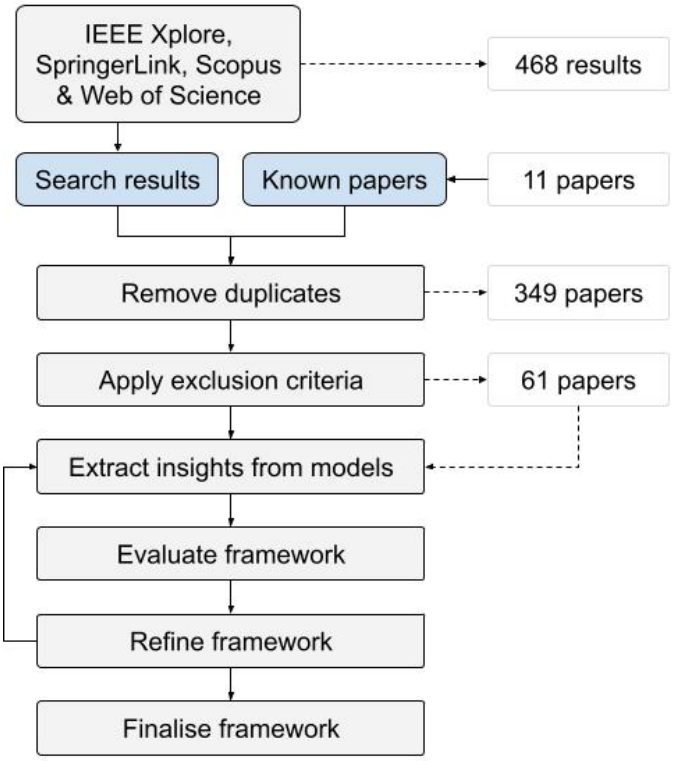


FIGURE 4.3: An illustration of the systematic literature review process employed in this study.

Database name	Results returned	Search options
IEEE Xplore	95	Searched from within 'All Metadata'
Scopus	149	Searched in the 'Computer Science' area
Springer link	24	Searched in the 'Computer Science' discipline
Web of Science	200	Searched in the 'Computer Science' area

TABLE 4.1: The number of results returned from the search from each database.

We considered only journal articles, rather than conference papers, in order to maintain a manageable volume of publications to consider and to ensure the highest level of quality of the papers analysed. The search query was executed on the title, abstract and keywords of papers in those search engines and - for practical reasons - considered only English-language publications. There were no limitations on the publication dates being considered.

The number of results returned by each of the scientific databases considered is shown in Table 4.1. Due to variations in the classifications of articles for each database and variations in the searchable fields available, Table 4.1 also mentions any search options which are specifically relevant to the searches carried out on each database. The complete configuration of each of these four searches is specified in Tables B.1 to B.4 of Appendix B.

As shown in 4.3, by combining the results of the four databases searched, we returned

349 unique papers, since a large number of the papers returned were duplicated across multiple databases. The exclusion criteria which had been defined were then applied to ensure that the papers returned from the searches were suitably relevant. At least one of the following relevance criteria must be met to be included:

1. Proposes a cyber security threat model
2. Proposes a cyber security threat modelling process
3. Proposes a cyber security threat modelling method

If at least one of the relevance criteria is met, then in addition all of the following requirements must be met:

1. Makes a contribution to cyber threat modelling
2. Is not a systematic review or survey paper
3. Is written in English
4. Is not a derivative paper

The remaining papers were manually screened and a further set of exclusion criteria were applied. In particular, articles were excluded if they:

1. Were privacy-based models
2. Merely apply an existing threat model or threat modelling process (such as STRIDE) to a particular IT system
3. Merely describe a framework
4. Primarily relate to modelling trust, privacy or compliance
5. Merely discuss threat modelling, but do relate to the conduct of threat modelling, such as those which evaluate existing threat models

However, threat models which relate primarily to modelling threats to cyber physical systems or to the social-technical aspects of cyber security are considered to be in scope.

Each of the main stages of the SLR process shown in Figure 4.3 resulted in a new collection of papers in our reference manager (Mendeley). At each stage these were exported to a respective CSV file to be imported into a spreadsheet to facilitate further steps required at each stage (i.e. de-duplication, applying exclusion criteria, conducting the

main analysis). Using this process, 61 papers were identified as passing the exclusion criteria and so are suitable to be subject to subsequent stages of analysis.

During the analysis of these papers, each of the threat modelling activities in our draft framework are assessed on the basis of whether they are:

1. Used *explicitly* (E),
2. Used *implicitly* (I),
3. *Unused* (U) but could, in principle, have been used,
4. *Could not have been used* (C), or whether it is
5. *Ambiguous* (A) as to whether they are described as being used in the article in question.

In addition to assessing each activity on this basis, the papers were also examined to identify any additional activities which are presented as being necessary components of any threat modelling process. These are to be used as candidates for new activities in the second round of evaluation.

The final collection of journal articles used in the SLR process is shown in Table B.5 of Appendix B and Figures B.1 to B.4 illustrate the analysis results for each article.

4.2.1 Findings and discussion first evaluation round

The summarised results of the first round of the systematic literature review-based evaluation are illustrated in Figure 4.4.

With reference to Figure 4.4, it is clear that the definition of a goal of the threat model or threat modelling process/method is typically provided explicitly in the corresponding article. Further some kind of characterisation of the IT system and identified threats are also generally provided explicitly. A core model is usually an implicit or explicit component of the threat model. While mitigations are discussed more often than not, most threat models do not explicitly discuss how controls are identified and characterised in the context of the threat model. This is likely to be due to the limited scope of some threat models in that they are often primarily concerned with the precursors to mitigating threats. However, since threat modelling is a process which is ultimately intended to determine a suitable threat response, we do consider these stages to be a necessary part of a complete reference framework for asset-based threat modelling.

Lastly, around half of the articles evaluated explicitly mentioned some kind of validation step (typically a case study, proof of concept, or expert review), and we assess

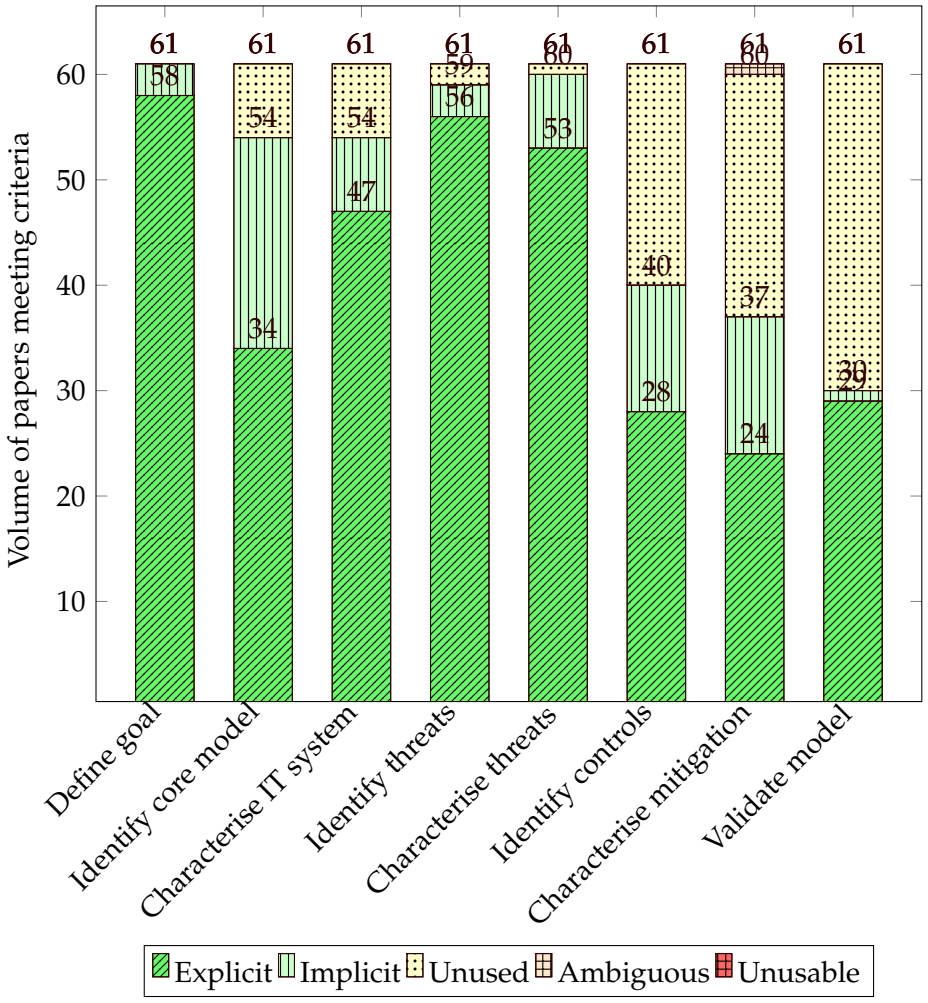


FIGURE 4.4: A stacked bar chart illustrating the frequency of each classification of each proposed threat modelling activity, as classified based on the identified threat modelling articles.

that all of them would benefit from being validated using an appropriate technique. Therefore, we also determine this step to be an appropriate component to include in our reference framework.

In addition to the quantitative insights discussed above, we also discuss several key qualitative insights identified throughout the first round of the SLR-based evaluation.

4.2.1.1 Definitions of key terms

An important observation from reading the selected papers is that several emphasised the importance of establishing clear definitions of key terms (Meszaros and Buchalcevova (2017), Hamad and Prevelakis (2020), Rak et al. (2022)). The tight relationship between clear definitions and the validity of the 'core model' is also apparent. For example, Wang and Guo (2009) present an Ontology for Vulnerability Management

which includes 'top level concepts' such as *vulnerability*, *IT product*, *attack* and *countermeasure* as those which are 'essential to characterise security vulnerabilities'. As such, the authors note that 'therefore, they should be defined ...as high-level concepts'. Each of these key words are given clear and explicit definitions in that publication.

Further, Herzog et al. (2007) state that an 'ontology can be used as an extensible dictionary of the domain of information security'. They note that the 'classic components of risk analysis' include assets, threats, vulnerabilities and countermeasures, and that they form 'core concepts' in their ontology of information security. They provide explicit definitions, not only of each concept, but also of the relationships between each of them, emphasising the importance of this step in the process of defining a core ontology. This is seen throughout the selected papers as an important role of ontologies, in '[defining] the basic terms and relations comprising the vocabulary of a topic area' (Gao et al. (2013)). In other papers, there is an explicit provision of definitions for such key terms, and an implicit assumption as to their importance in properly specifying a core model (Jbair et al. (2022)). This analysis also led to the observation that papers with clear definitions are likely to have a more clearly identifiable 'core model'.

4.2.1.2 Characterising threats and controls

It was also observed that where studies had not explicitly characterised the IT system concerned, the identified threats were generally only characterised in terms of their behaviour (their effect on the system) or agency (who presents the threat and why) rather than their composition (the physical or logical components of the system, and their configuration, which are required for the threat to become manifest).

Attack-tree-based models, for example, typically did not describe the composition of threats (Potteiger et al. (2016)). Where they did, this composition was often in terms of the attackers and motivation necessary to realise a threat. In the context of software-based models, the composition was typically expressed using data flow diagrams (Egoshin et al. (2020)) or activity diagrams (Li et al. (2014)).

However, in the case of threat models that were more distinctly asset-based, as is our primary interest, these typically contained a detailed characterisation of the underlying IT system being considered. As such, the composition of cyber threats was often referenced, in addition to (or as an alternative to) reference to the mere behaviour of cyber threats (Joshi et al. (2020), Xiong et al. (2018)). This was often achieved by associating a particular threat with a set/pattern of assets (and asset properties) that are required for the threat to be considered to exist. As such, it appears that both the behaviour and the composition of threats *may* be characterised in any given asset-based threat model. Likewise, in asset-based threat models, threat countermeasures were often defined in

terms of the asset which they target, in addition to the action which they perform on the IT system.

Further to the primary insights discussed above, additional features of some of the evaluated articles are as follows:

- The majority of the core models had significant similarity in the composition and relationships between concepts of their core model, typically using the concepts of *vulnerability*, *mitigation*, *asset* and *threat*.
- Several asset-based models used data flow diagrams (DFDs) in order to represent their core model and to illustrate particular IT systems, however this is never considered a requirement of asset-based modelling.
- Several articles provide considerable guidance to the reader in how to apply the threat model or threat modelling process they describe. We do not consider this to be a specific 'activity' of threat modelling, but simply a beneficial aspect of such processes.
- Some articles used reference architectures from other papers in order to support the utility or validity of their models.

While these additional points of discussion add valuable context to our evaluation, we do not consider them to imply that additional components need to be added to the reference framework.

4.3 Amendments to the draft framework

In response to the findings from the first round of evaluation, discussed in the section above, a number of changes have been made to the threat modelling activities which will be part of the generic threat modelling process:

1. An activity was added for the relevant terms to be defined (a *define terms* activity). This was based on the observation that many of the papers analysed included an explicit stage in the threat modelling process for clarifying the precise meaning of the key concepts which were relevant to their threat modelling process. The identification of the core model has also been amended to clarify it should be *extracted* from the definition of the key terms and the goal.
2. The *characterise threats* activity has been divided into two complementary stages. These two stages comprise one activity for *characterising threat behaviour* and another activity for *characterising threat composition*.

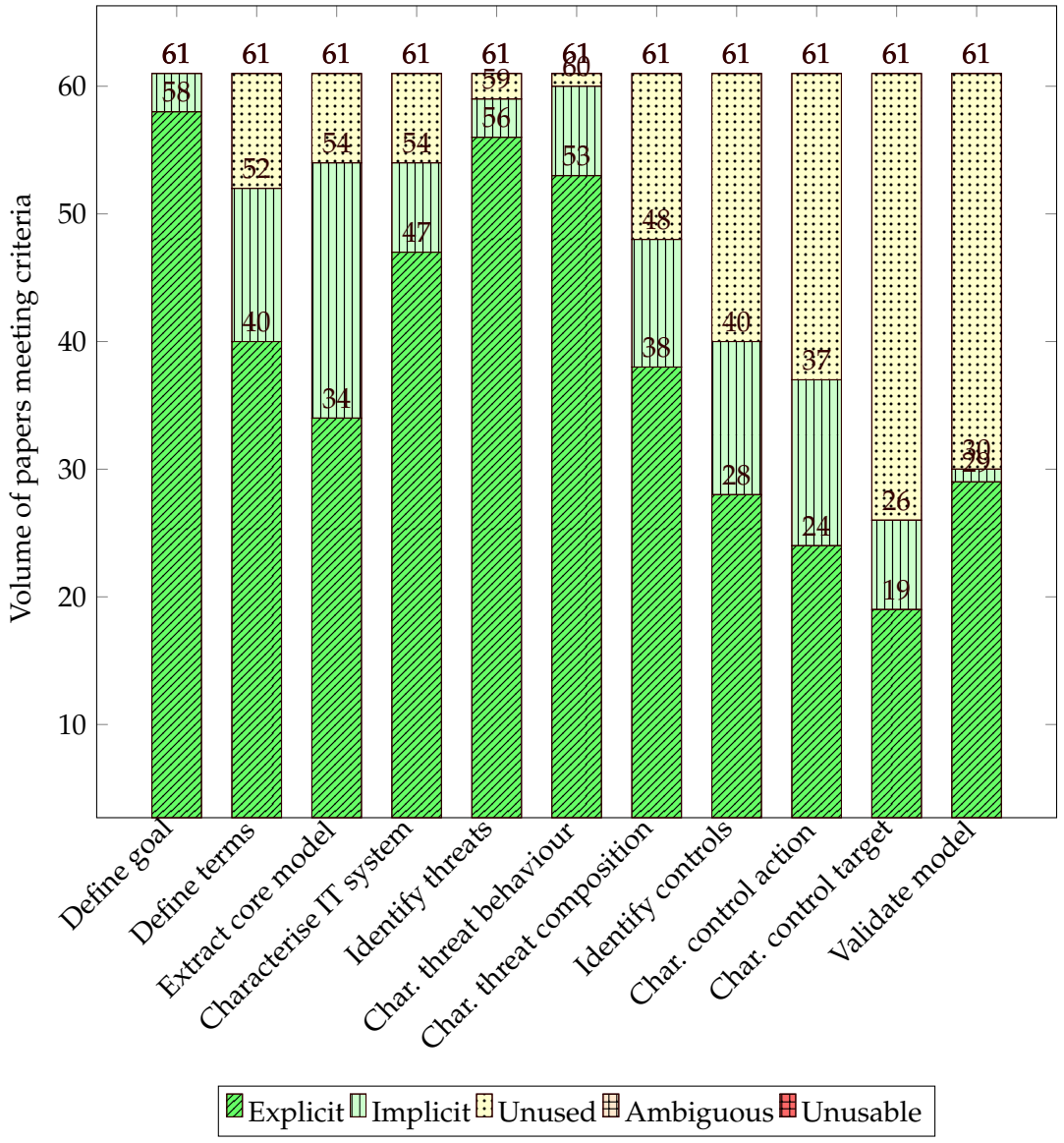


FIGURE 4.5: A stacked bar chart illustrating the frequency of each classification of each of the amended threat modelling activities, as classified based on the identified threat modelling articles.

3. The *characterise threat mitigation* activity has also been separated into two complementary stages. These two stages comprise an activity to *characterise the control action* (i.e. behaviour) and another activity to *characterise the control target* (i.e. a relation of the countermeasure to its associated components/assets in the IT system).

On the basis of these refinements, the SLR-based evaluation process was adjusted and the amended or added activities have been evaluated, with results illustrated in Figure 4.5.

4.4 Key findings and discussion

This analysis process sought to analyse the sufficiency of the draft threat modelling framework at characterising the activities which constitute the basic components of a rigorous asset-based threat modelling process.

As shown in Figure 4.5, our second round of evaluation identified the relevance of each of the new or amended threat modelling activities to the corpus of articles identified in the SLR. Of particular note, the vast majority of articles explicitly or implicitly define key terms used in their model. In respect of the threat characterisation, it is more common to seek to characterise their behaviour rather than their composition. This may be because the technical impact (i.e. behaviour) of a threat is more directly accessible than how that threat arises from the composition of the underlying IT system, and is therefore an easier aspect to focus on in a threat model. However both of these activities are very common and are compatible with all types threat models.

With regards to characterising controls, it is more common for them to be characterised in terms of their action on the overall system. However, for those articles which did discuss control characterisation, the distinct majority did so in terms of the target of the controls too.

It may be considered appropriate to infer the relative importance of each activity based on the extent to which the stages are explicit or only implicitly mentioned. However, we suggest that this is merely a reflection of the relative scope and robustness of the studies being evaluated. For example, only 34 out of 61 papers explicitly presented what might be considered to be a 'core model', however rather than diminishing the importance of this stage to asset-based threat modelling, we suggest this merely highlights the limited scope, or the subjective nature of several asset-based threat models, and the need for a more robust approach to threat model development.

In short, all of the stages of the amended set of activities are at the very least compatible with all of the threat models in the evaluated articles. Most are explicitly mentioned, while many more are referred to implicitly. This perspective is assessed to provide substantial and robust quantitative and qualitative validation of both the relevance and sufficiency of these set of activities in representing a generic asset-based threat modelling process. Since no further additional activities were clearly identified in the second SLR round as potential candidates we did not consider a third SLR round to be required.

It should further be highlighted that this process is based on an explicitly-defined method based on a systematic literature view. The method itself has wide credibility and usage in the field and the specific sources, queries, constraints and SLR questions used in this Chapter have all been documented. Accordingly, this experiment is readily repeatable and extensible to those wishing to further validate our results.

4.4.1 Limitations

As described above, the SLR used a large volume of academic publications from a diverse range of sources. However, there are many more publications which could have been included. The process of identifying, filtering and analysing papers as part of a systematic literature review is a time-consuming process, and there would not have been a sufficiently enhanced benefit in widening the sources used. The publications analysed were drawn from reputable search engines and since only journal articles were searched for we were able to more efficiently validate the quality of the articles. Although we could have included conference papers and workshop proceedings, it would have been more complex to guarantee the right level of quality and would have left the process with too many papers to realistically manage.

Further, the volume of papers analysed is considered to be sufficiently large to provide statistical significance to the results and therefore to the in-principle validity of each of the activities. As such, we consider the articles reviewed to provide a high-quality knowledge base of cyber threat modelling, suitable for confirming the primary research objective described by RQ1 in Chapter 3.

4.5 Asset-based threat modelling process

On the basis of the asset-based threat modelling activities established and validated in this Chapter, we have translated these activities into a visual representation in the form of a reference framework. We refer to this framework as the *Reference Framework for Asset-based Threat Modelling* (*ReFAThM*), and is illustrated in Figure 4.6. In that Figure, the grey components indicate requirements or dependencies of the framework, including and required data sources. The blue components indicate the activities which this chapter has identified as being necessary and sufficient activities of a generic asset-based threat modelling process. The green components indicate outputs of a corresponding threat modelling process.

This reference framework does not add any significant new information and thus does not entail any significant new assumptions, thereby being sufficiently supported in its entirety by the above-mentioned findings. However, its representation in this form enables it to be used more easily for practical purposes, including in evaluating the completeness of existing asset-based threat modelling processes and methods, evaluating the robustness of existing asset-based threat models and for guiding the process of developing new such models, processes and methods in a rigorous and structured manner. Indeed, the findings of this chapter, represented in this reference framework, are used and referred to in subsequent chapters of this thesis, in order to improve their relevance and academic validity.

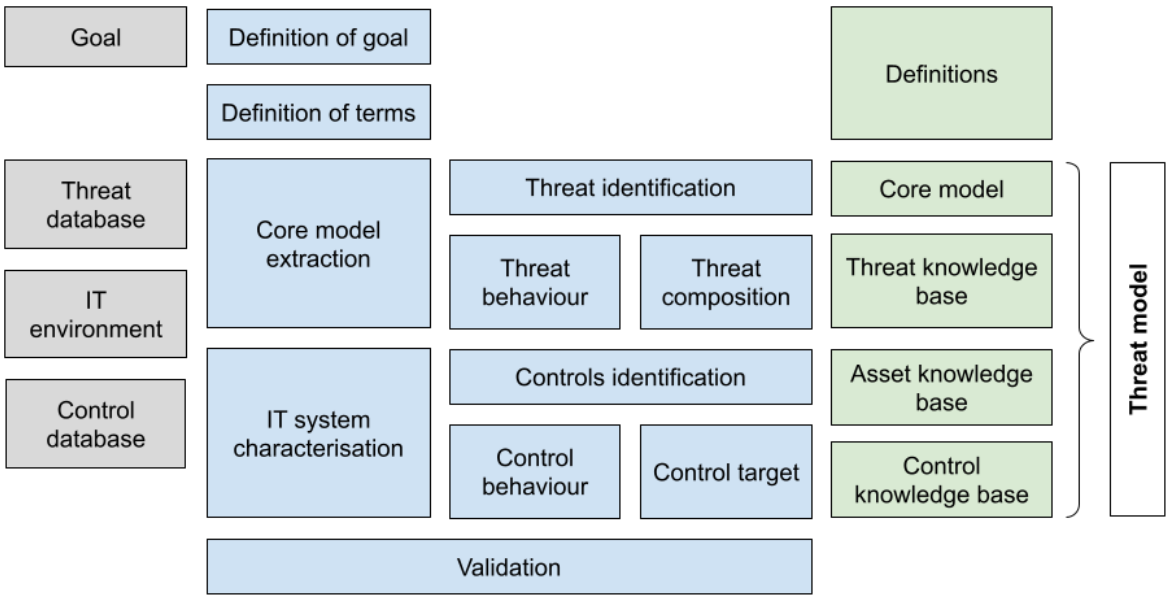


FIGURE 4.6: A reference framework illustrating the asset-based threat modelling process as established by this research activity.

The definitions of the final asset-based threat modelling activities are defined as follows:

1. **Define goal:** the purpose of the threat model, or threat modelling method or process, should be clearly defined.

2. **Define terms:** the key terms which relate to the goal of the threat model should be clearly defined.

3. **Extract the core model:** the essential concepts necessary to characterise a threat must be identified and extracted from the goal and key terms definition, and the relationships between them established.

4. **Characterise IT system:** the essential nature of the IT system must be expressed, such as, in terms of the flow of data, the IT assets involved, or the properties of the IT environment within which threats may emerge.

5. **Identify threats:** the name or other identifier of at least one threat which is relevant to the IT system must be provided.

6. **Characterise threat behaviour:** the essential nature of the threat must be expressed with reasonable detail and clarity, in terms of its technical impact or behaviour on the IT system.

7. **Characterise threat composition:** the essential nature of the threat must be expressed with reasonable detail and clarity, in terms of how it is manifested with reference to the assets which comprise the IT system.

8. **Identify controls:** the name or other identifier of at least one mitigation to one of the aforementioned identified threats must be provided.

9. **Characterise control action:** the essential nature of the aforementioned control(s) must be expressed with reasonable detail and clarity, in terms of the alteration made to the composition or configuration of one or more system assets in mitigating a corresponding threat.

10. **Characterise control target:** the essential nature of the aforementioned control(s) must be expressed with reasonable detail and clarity, in terms of the specific system asset(s) which the alteration must target in order to mitigate its corresponding threat.

11. **Validation:** the rationale underpinning the implementation of the threat model, or threat modelling process, should be validated using a suitable technique, such as using a case study, expert review, semantic modelling, ontology engineering or using formal methods.

4.6 Summary

In this Chapter, we have highlighted the limitations in existing literature related to asset-based threat modelling. In particular, there is currently no methodological rational offered to guide the development of asset-based threat modelling activities. This means that such efforts are currently fraught with subjective design decisions and with limitations, which are not readily obvious to the model developer.

In response to this limitation, we have identified the key activities involved in a generalised asset-based threat modelling process and have validated their coverage of and relevance to the threat modelling process using a systematic literature review. These activities have been represented in the form of a structured reference framework for asset-based threat modelling which we have called 'ReFAThM'. This framework serves as a tool which can be used throughout the lifecycle of asset-based threat model design, implementation, application and refinement in order to underpin a robust qualitative evaluation of its relevance and validity.

Chapter 5

Mapping the Threat Landscape with Topic Modelling

This chapter presents a method for developing a concise cyber threat knowledge base from a large corpus of cyber vulnerability information. It is based on joint work with Tsang et al. with whom we have published some of its main results in Tsang et al. (2024).

The background and justification of the methodological approach taken in this chapter is described in Chapter 3.

5.1 Data source and enhancements

In order to characterise a knowledge domain using topic modelling we first identified the Common Weakness Enumeration (CWE) dataset as our primary data source. We identified 503 CWE entries comprising the full 399 CWEs related to software development and a further 104 to hardware design. These two categories were included in order to establish a broad level of coverage of the threat landscape. The CWE dataset offers a similarly broad coverage to the CVE dataset, which was also a viable option, however the CWE dataset offers a more suitable semantic range for conducting language-based clustering at a higher level of abstraction.

Although CWE entries are somewhat structured and contain certain threat attributes, we wanted to maintain control over which threat attributes we sought to analyse. As such, we conducted a manual clustering of the set of attributes comprised within a number of existing threat or vulnerability taxonomies. These include the CVE, CWE, CAPEC and ATT&CK taxonomies as well as the ReFAThM framework described in Chapter 4. We also used a large language model (GPT-3.5) to enumerate the attributes of a cyber threat to further widen the scope of considered threat attributes.

Attribute	GPT-3.5	CVE	CWE	CAPEC	ATT&CK	ReFAThM	Total
Name	Yes	Yes	Yes	Yes	Yes	Yes	**6**
ID		Yes	Yes	Yes	Yes		**4**
Description	Yes	Yes	Yes	Yes	Yes		**5**
Vulnerability	Yes	Yes		Yes		Yes	**4**
Method	Yes				Yes		**2**
Technical impact			Yes	Yes			**2**
Security properties	Yes		Yes	Yes			**3**
Severity	Yes			Yes			**2**
Likelihood			Yes	Yes			**2**
Asset	Yes		Yes		Yes	Yes	**4**
Attack vector	Yes		Yes	Yes			**3**
Attacker type	Yes						**1**
Attacker motive	Yes						**1**
Controls	Yes		Yes	Yes	Yes	Yes	**5**
Detection			Yes		Yes		**2**

TABLE 5.1: A table illustrating which of the 6 threat attribute sources includes each of the attributes considered. These threat attributes include 12 primary threat attributes, alongside a name, ID and description field.

Many of the attributes of each taxonomy used different terms to describe similar attributes. These synonyms were clustered into the same threat attribute. For example, the following terms were all considered to be synonyms, or sub-categories, of our term 'asset': target, system, software, affected asset, applicable platforms, technologies and languages. The final list of attributes which were arrived at, along with the threat taxonomies which refer to them, are shown in Table 5.1. It is these specific attributes which we use in the following stages to perform threat characterisation using feature importance analysis.

Since none of the taxonomies above comprised descriptions for all threat attributes (indeed, in the case of the CWE dataset, even the fields which it comprises are often incomplete or omitted entirely) it was necessary to construct our own multi-dimensional threat description corresponding to each CWE entry (i.e. a description of each threat attribute for each CWE entry). To do so, we considered a number of options. One such option was to simply reduce the number of threat attributes considered and use only the threat attributes contained in the raw CWE entry or the corresponding CAPEC entry. This was considered to limit the scope of threat attributes too much. Another option was to use a qualitative process involving the use of subject-matter experts. This would have required using a large number of human subjects creating a vast quantity of human-readable content, alongside a cross-validation process. As such, this was deemed to be too costly in time and resources.

We therefore identified that GPT-3.5, which had just been released to the public at the time this work was taking shape, could be used to generate text-based descriptions of each of the 12 threat attributes, for each CWE entry. The threat attributes which were

selected were: vulnerability, method, technical impact, security properties (those affected by the threat), severity, likelihood, asset (relevant to the threat), attack vector(s), attacker type(s), attacker motive(s), controls and detection (detection methods). GPT-3.5 was used to generate human-readable descriptions for each of these attributes in order to construct a complete dataset for further analysis.

The prompt used to instruct GPT-3.5 was the following:

Use what you know about this CWE and the description provided to describe the following attributes of this threat for me: the vulnerability, method, technical impact, security properties affected, severity, likelihood, relevant assets, the attack vector(s), the attacker type(s), the attacker motive(s), relevant cyber controls/countermeasures, and detection methods. Please use full sentences for each attribute.

This prompt was asked in respect of each of the 503 CWE entries, resulting in a dataset too large to reproduce in an appendix. However an example of the output for one CWE entry is given in Table 5.2. This dataset is then used in the topic modelling stage to cluster each entry and to identify the inherent taxonomic structure of this corpus of threat information.

5.2 Model evaluation

Although each threat description (represented by the 12 LLM-synthesised simulated threat descriptions, derived from a corresponding CWE entry) are unique and relate to distinct information-system weaknesses, they are all part of the narrow field of cyber security. Therefore, for the reasons mentioned in Chapter 3, we selected BERTopic as a flexible, modular and high-performance Python library for using the BERT transformer-based language model for conducting topic modelling. In order to evaluate the comparative merits of each configuration of this topic model, we developed the process illustrated in Figure 5.1.

5.2.1 Threat Description Embeddings

Beginning with the LLM-synthesised text-based descriptions of each of the 12 threat attributes corresponding to each CWE entry, we first identified two potential approaches for generating the embeddings. A Unified Document Approach (UNI) would involve combining the text of each attribute to treat them as a single document to be clustered. Alternatively, an attribute-specific approach (ATT) would treat each attribute separately by generating separate embeddings for each, to be combined prior to dimensionality reduction. We were curious as to whether this would yield differences in clustering performance.

Threat attribute	Description generated by GPT-3.5
Vulnerability	The vulnerability in this case is the lack of proper validation of XML against the appropriate schema.
Method	The method used by the attacker is to provide malicious XML to the product from an untrusted source.
Technical impact	The technical impact of this vulnerability is that the product may process the malicious XML, leading to various consequences such as data corruption, denial of service, or even remote code execution.
Security properties	The security properties affected by this vulnerability are confidentiality, integrity, and availability.
Severity	The severity of this vulnerability depends on the specific consequences that may result from processing the malicious XML. It could range from low to critical.
Likelihood	The likelihood of this vulnerability being exploited depends on the product's popularity and the ease of providing malicious XML to it.
Asset(s)	The relevant assets in this case are the product that accepts XML from an untrusted source and any data or systems that may be affected by the consequences of processing malicious XML.
Attack vector	The attack vector in this case is the untrusted source providing malicious XML to the product.
Attacker type	The attacker type in this case could be anyone with the ability to provide malicious XML to the product, including insiders or external attackers.
Attacker motive	The attacker's motive could be to gain unauthorized access to sensitive data, disrupt the product's functionality, or cause other malicious consequences.
Controls	Relevant cyber controls/countermeasures include implementing proper XML validation against the appropriate schema, restricting access to the product, and monitoring for suspicious activity.
Detection	Detection methods include monitoring for unusual XML processing behavior, analyzing logs for signs of malicious XML, and implementing intrusion detection systems.

TABLE 5.2: A table illustrating the descriptions for each threat attribute for CWE-112 as generated by GPT 3.5

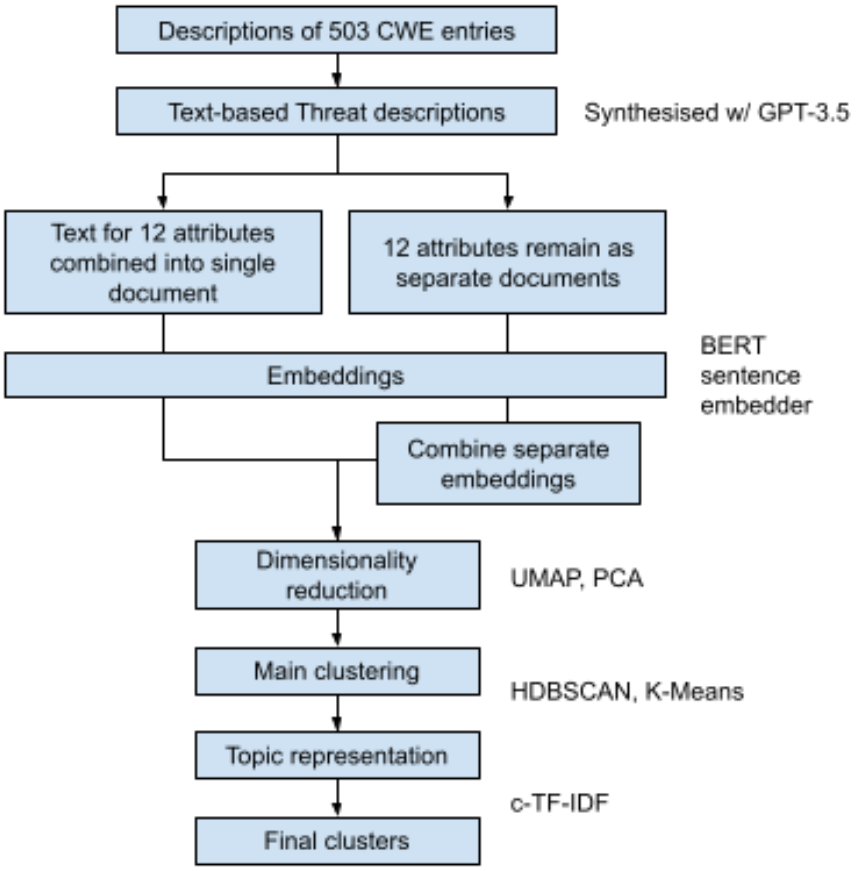


FIGURE 5.1: An illustration of the Topic Model evaluation process.

Embeddings are often used in NLP to represent text-based data as numerical vectors in order to facilitate clustering using conventional techniques. This means that two documents with similar semantic meanings are more likely to have smaller distance vectors between their embeddings than documents which are more dissimilar, thus enabling documents to be clustered based on their meaning.

A range of embedding models are compatible with BERTopic, however we used the SBERT SentenceTransformer Python library for its range of pre-trained general-purpose models which had all been extensively evaluated for their quality in conducting sentence embedding.

Among the pre-trained models available, we used the *"all-MiniLM-L6-v2"* sentence-transformers model from *Hugging Face*. This model maps documents to a 384-dimensional vector space and is designed as a general-purpose model for clustering. It offers versatility and a suitable performance-efficiency trade-off, and was trained with an extensive training dataset. Since our dataset of cyber threat data is from a relatively narrow and ubiquitous field, and uses short paragraphs, this model was deemed likely to encode the nuanced differences between dataset entries most effectively.

5.2.2 Dimensionality reduction

In order to overcome a variety of performance-related challenges with clustering, it is common to perform dimensionality reduction on the samples being clustered. Embedding samples using our unified document approach comprise the 384 dimensions used by the sentence embedding model. Using the attribute-specific approach, the 12 individual attribute embeddings are combined, resulting in samples with 4,608 dimensions. Therefore, due to the large dimensionality, there is a particular need in our use-case to reduce sample noise and over-fitting, as well as to improve the computational efficiency of performing clustering.

Two methods of dimensionality reduction were employed in this study:

1. **UMAP**: Uniform Manifold Approximation and Projection is used by default in BERTopic. Its main strength for our purposes is that UMAP is more suitable for preserving both local and global structure (i.e. the relationships between both adjacent and dissimilar samples). While it is more computationally complex, UMAP is generally more suitable for preserving non-linear relationships, potentially enhancing clustering accuracy.
2. **PCA**: Principle Component Analysis, however, is more likely to retain the global structure, but lose the local relationships, since it focuses on maximising variance along the principle components.

Since PCA is a common technique, we used it to provide a reference point, enabling us to compare BERTopic's default of UMAP to it.

5.2.3 Clustering

Topic modelling uses clustering in order to group samples with similar semantic meaning into clusters, corresponding to distinct topics within the overall knowledge domain represented by the original dataset. We evaluated two clustering algorithms in our study: HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) and K-Means.

HDBSCAN is the default method used by BERTopic and offers a number of advantages over K-Means:

- It uses density-based clustering, meaning clusters are identified based on the density of samples within the feature space. In the case of text-based data, samples in the same topic are likely to use very similar, or the same, words corresponding to more similar embeddings. This may correspond to more dense clusters

where the text samples are similar and more sparse clusters when text samples are dissimilar. In contrast, K-Means uses centroid-based clustering.

- It can capture clusters of varying sizes more effectively, making it more likely to properly characterise both large topics as well as more niche ones, rather than defining cluster boundaries arbitrarily based on size. K-Means however, generally identifies clusters of similar size and shape which is less likely to be appropriate for modelling topics within text-based datasets.
- Unlike K-Means, HDBSCAN determines the number of clusters from the data rather than needing to specify them in advance, as with K-Means.
- It is able to identify outliers which do not fit within any discernible cluster. However, this could mean too many outliers are identified. K-Means, by contrast, does not accommodate outliers.
- Unlike K-Means, HDBSCAN is a hierarchical clustering algorithm, meaning it can represent multiple layers of clusters. This will later be helpful when performing cluster merging, making it easier to make informed decisions about refining and consolidating identified topics.

Due to its well-established role as a reference point, we have used K-Means as our baseline algorithm from which we have compared HDBSCAN.

5.2.4 Topic representation

In the context of topic modelling, topic representation is the process of characterising the topic which each cluster corresponds to. Typically, this is conducted by extracting the key terms which correspond to the central theme within each topic. The method used by BERTopic to perform this process is called class-based Term Frequency-Inverse Document Frequency (c-TF-IDF). This is a variation of the traditional TF-IDF weighting scheme which weights key terms on the basis of their frequency within constituent documents (Term-Frequency) and their rarity across all documents (Inverse Document Frequency).

Class-based TF-IDF additionally incorporates the discriminative power of each term with respect to different classes into that weighting (i.e. giving more weight to terms which are relevant to specific classes and not relevant to others), making it more suitable for text-based classification tasks. Using c-TF-IDF, the main theme of each cluster is characterised by identifying the top N words with the highest c-TF-IDF scores which, in principle, represent the essential terms which distinguish one cluster from another.

BERTopic only offers c-TF-IDF as its topic representation algorithm, however it enables the user to specify the number N of terms to extract. We set N to 20 in order to provide

plenty of terms for enabling us to characterise the latent themes within each cluster, however the terms are given in order of their total weight, so earlier terms were given more prominence when characterising the latent theme from the terms.

5.2.5 Model selection and hyperparameter tuning

As described above, with reference to Figure 5.1, our model evaluation choices comprise two pre-handling techniques (UNI and ATT), two dimensionality reduction methods (UMAP and PCA) and two clustering algorithms (HDBSCAN and K-Means), resulting in eight possible model configurations. However, in addition to varying these options, the dimensionality reduction (for both UMAP and PCA) and clustering algorithms (for both HDBSCAN and K-Means) both involved the evaluation of a range of model hyperparameters which needed to be configured by the user. The process of tuning these hyperparameters is an optimisation problem with the goal of maximising the model's performance.

Our primary consideration for hyperparameter tuning was to optimise both the extent of dimensionality reduction applied and the type of clustering adopted. Due to the number of hyperparameters being considered, the computational complexity of this hyperparamter tuning problem would ordinarily have been excessive if every permutation had been evaluated. Since these two factors are inter-dependent, our tuning strategy focused on adjusting either one or two hyperparameters from each method at a time, and retaining default values for the others.

We adopted the use of two types of evaluation metrics for assessing the relative performance of each model configuration[1] and their corresponding hyperparameter settings[2]:

1. General clustering metrics, which are applied to the embeddings prior to dimensionality reduction and are paired with cluster labels, and
2. Topic modelling-specific metrics, which are applied to the key terms generated for each cluster.

The general clustering metrics used are the Silhouette Method (Rousseeuw (1987)) and the Calinski-Harabasz (CH) Index (Caliński and Harabasz (1974)). The topic modelling-specific metrics used are topic diversity (Dieng et al. (2020)) and coherence scores (Röder et al. (2015)). These techniques are described in the following:

- The **Silhouette Method** measures how similar an object is to its own cluster, by comparison to other clusters. The Silhouette score is made up of Silhouette values

[1] In this case, a model configuration refers to a specific combination of data pre-handling, dimensionality reduction and main clustering approaches

[2] The specific parameterised variation of a given model configuration

for each sample. These range from -1 to 1 and are a measure of how closely related that sample is to other data points in its own cluster and how poorly it is matched to neighbouring clusters. For example, a value close to 1 indicates it is well within its own cluster and far from other clusters; a value close to -1 indicates it is closer to points in a neighbouring cluster than its own, and a value of 0 suggests it is equidistant from its own cluster and an adjacent cluster.

- The **Calinski-Harabasz (CH) Index**, also known as the variance ratio criterion, is used to measure the quality of clustering. It is calculated based on the ratio of the sum of inter-cluster dispersion and of intra-cluster dispersion for all clusters. The higher the CH Index, the better the clustering.

- **Topic Diversity** is a measure of the distinctiveness of words within a topic, relative to words in other topics. A high diversity score suggests that the topics cover different concepts or themes within the knowledge domain covered by the dataset, implying there is minimal overlap between key terms among adjacent topics. This helps to avoid creating topics which represent similar themes.

- **Topic Coherence** is a measure of semantic similarity of key terms *within* a given topic. A high coherence score corresponds to a topic in which its top words have strong semantic similarity indicating the topic represents a distinct concept or theme in the overall knowledge domain. We used three techniques to measure coherence: Cv (Röder et al. (2015)), UMass (Mimno et al. (2011)) and NPMI (Bouma (2009)).

In addition to the metrics described above, we also incorporated a measure of qualitative evaluation of the quality of each model. In particular, we considered the presence of excessively dominant clusters to suggest sub-optimal topic modelling. Further, where there was an excess of outliers (as can arise with the HDBSCAN method), this was interpreted as indicating a weakness in the quality of the topic modelling. Further, it is considered relevant to consider that models with optimal quantitative metrics may not always correspond to clustering which matches a qualitative assessment. Therefore, it is considered appropriate to interpret the objective metrics through the lens of a qualitative assessment to identify the most suitable topic modelling approach to use. This is justifiable based on the inherent constraints of using quantitative techniques to characterise a subjective knowledge domain and therefore requires that any statistical observations are contextually relevant through the use of qualitative validation.

Therefore, selecting a final model configuration involved a two-step process:

1. **Quantitative Evaluation**: Propose a shortlist comprising one or two parameterised model variation for each of the eight model configuration using the quantitative metrics described above.

- An exhaustive search of all model configuration and hyperparameter settings was not pragmatic due to the large volume of hyperparameters and the complex inter-dependencies between each parameter value. Therefore, we developed a set of criteria for short-listing candidate hyperparameter settings:
  (a) The hyperparameter settings must consistently correspond to satisfactory scores (ideally within the top 50%) across the majority of the metrics.
  (b) None of the metrics should be in the bottom 25% by comparison to its peers.
  (c) Hyperparameter settings resulting in a pronounced dominant cluster were excluded.
  (d) Hyperparameter settings resulting in an excess of outliers were excluded.

2. **Qualitative Evaluation** Conduct a final model and parameter selection by augmenting the quantitative metrics of the shortlisted models with a qualitative assessment of the quality of the resultant clusters from each model.

   - This was based a qualitative assessment of the extent to which the resultant topics corresponded to identifiable threats recognised in the field of cyber security.
   - It was also based on a qualitative evaluation of the metrics, such as the number of clusters, the sizes of the largest clusters and the number of outliers.

5.2.6 Evaluation of hyperparameter and model configurations

During the initial stages of implementing the evaluation process described above, we observed that models combining PCA with HDBSCAN resulted either in a large number of outliers or in a dominant cluster. Based on the criteria above, this implied that that particular configuration was not suitable and was therefore excluded.

With reference to the remaining model configurations illustrated in Table 5.3, we can make a number of further observations:

- There were far fewer numbers of qualified hyperparameter setting combinations when PCA was used for dimensionality reduction rather than UMAP, indicating that UMAP preserves a richer representation of the structure of embeddings than PCA.
- Clustering using HDBSCAN resulted in more clusters than using K-Means. It also resulted in greater variability in the size of clusters by comparison to K-Means.

- HDBSCAN generated a considerable number of outliers. However, at approximately 20% of the total number of samples, this was considered to be acceptable. K-Means does not accommodate outliers since all samples must be allocated a cluster.

Tables 5.4 and 5.5 provide illustrations of the 10 model and hyperparameter configurations which were shortlisted based on the quantitative analysis. This analysis was designed to evaluate the general effectiveness of a wide variety of potential model configurations, rather than definitively identify the best-performing configuration. As shown, there are a broad range of divergent performances among each of the down-selected configurations.

As expected, there is an inverse relationship between topic diversity and coherence. Topic diversity is of particular relevance in our case, since it indicates that clusters are distinct from each other and therefore that there is greater difference in their respective semantic meanings. However, in evaluating and selecting a model configuration, it is important to optimise their collective performance rather than just being based on diversity. For instance, topic coherence also indicates that the topic words in each cluster are closely related, implying that the cluster does, in fact, represent a single coherent theme. Since most topic words for all clusters relate to the field of cyber security, the C_v coherence score may be less significant for the purposes of evaluation than diversity.

Generally, model configurations employing the unified document approach (UNI) performed better than those using the attribute-specific approach (ATT) with respect to generic model evaluation metrics, such as the Silhouette score. The higher dimensionality of the ATT approach probably had a negative impact on the resulting clustering metrics, since it may have inhibited the performance of the dimensionality reduction. However, in terms of the two topic modelling evaluation metrics, both pre-handling approaches perform in a similar manner.

As seen in Tables 5.4 and 5.5, model configurations that used both UMAP (UMP) and HDBSCAN (HDB) together had a notably high topic diversity. The UNI+UMP+HDB model configuration was therefore selected based on its strong diversity score, and superior general evaluation metrics. In particular, the model configuration with ID 4 was chosen in preference to model 3, due to its slightly smaller average cluster size. This model configuration is deemed to offer suitable topic diversity and coherence, indicating that the clusters represent significant clusters of latent meaning which are valid for use in later stages of this research.

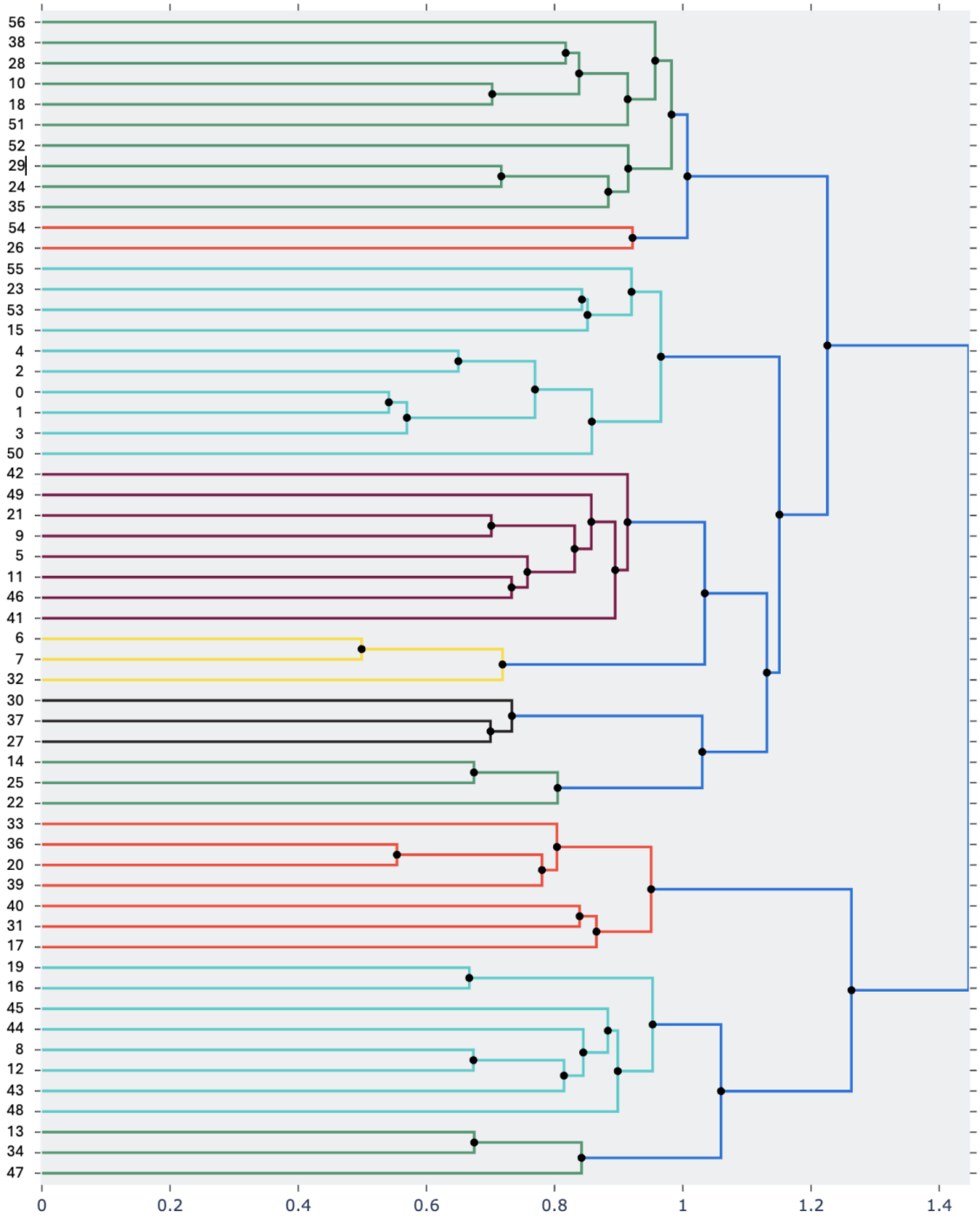


FIGURE 5.2: An dendrogram showing the hierarchical topic clusters using the model configuration ID 4 (as per Table 5.4).

5.3 Topic merging and final threat database

As seen in Table 5.4, the selected model configuration (ID 4) resulted in 57 clusters. These 57 clusters, along with their representative documents (i.e. CWE entries) and representative keywords (both as provided by SBERT) are shown in Table C.1 of Appendix C. Since many of these clusters comprised closely-related sets of keywords, we

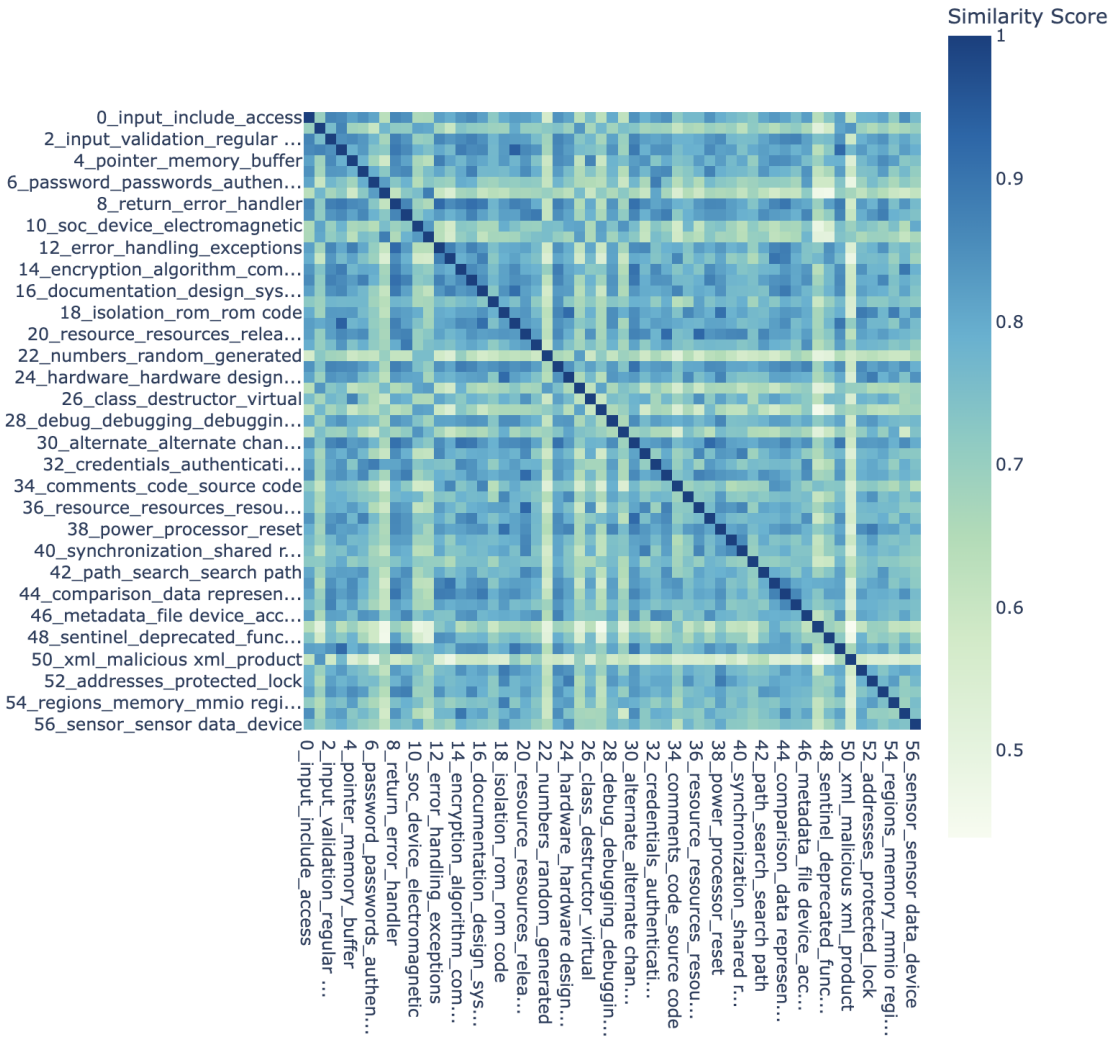


FIGURE 5.3: An illustration of the similarity matrix using the model configuration ID 4 (as per Table 5.4).

Model Config.[3]	Tuned Hyperparameters	No. Qualified[4]	Clusters[5]	Outliers	Smallest Cluster Size	Largest Cluster Size	Av. Cluster Size
ATT+UMP+HDB	n_neighbours, n_components	53	46-83(63.1)	55-112(86.2)	2-3(2.0)	15-63(38.0)	5.2-9.1(6.7)
UNI+UMP+HDB	n_neighbours, n_components	96	48-89(62.7)	54-149(105.3)	2	14-65(31.7	4.8-8.3(6.4)
ATT+UMP+KMS	n_neighbours, n_components, n_clusters	96	19	0	2-18(9.3)	41-74(49.2)	26.5
UNI+UMP+KMS	n_neighbours, n_components, n_clusters	96	10	0	6-23(20.9)	66-114(76.0)	50.3
ATT+PCA+KMS	n_components, n_clusters	14	28	0	1-6(3.9)	32-59(40.2)	18
UNI+PCA+KMS	n_components, n_clusters	10	7	0	21-39(28.9)	102-141(120.4)	71.9

TABLE 5.3: General comparison of model configurations

[3]In the abbreviations used in this column, ATT refers to the attribute-specific approach to pre-handling; UMP refers to UMAP; HDB refers to HDBSCAN; UNI refers to the unified document approach to pre-handling, and KMS refers to K-Means.

[4]This refers to the number of hyperparameter setting combinations which met the down-selection criteria.

[5]Cell formatted as min-max(mean).

ID	Model Config.	n_clusters (K)	n_neighbours	n_components	Clusters	Outliers	Min cluster size	Max cluster size	Avg cluster size
1	ATT+UMP+HDB		15	20	55	83	2	43	7.6
2	ATT+UMP+HDB		15	60	52	84	2	44	8.1
3	UNI+UMP+HDB		25	55	55	135	2	35	6.7
4	UNI+UMP+HDB		30	35	57	136	2	35	6.4
5	ATT+UMP+KMS	19	15	30	19	0	11	48	26.5
6	ATT+UMP+KMS	19	20	45	19	0	12	42	26.5
7	UNI+UMP+KMS	10	25	35	10	0	23	69	50.3
8	UNI+UMP+KMS	10	25	55	10	0	22	73	50.3
9	ATT+PCA+KMS	28		90	28	0	6	37	18
10	UNI+PCA+KMS	7		70	7	0	28	120	71.9

TABLE 5.4: Comparison of finalist hyperparameter setting combinations

ID	Model Config.	Silhouette	CH Index	Diversity	Cv	Umass	UCI	NPMI
1	ATT+UMP+HDB	0.0457	4.2512	0.8145	0.3927	-6.951	-6.3721	-0.1315
2	ATT+UMP+HDB	0.0461	4.3232	0.7827	0.4017	-6.1774	-5.9168	-0.1227
3	UNI+UMP+HDB	0.0635	5.4099	0.8455	0.3969	-6.9475	-6.2648	-0.1227
4	UNI+UMP+HDB	0.0725	5.4661	0.8211	0.3915	-6.5121	-6.3782	-0.1299
5	ATT+UMP+KMS	0.0328	8.3167	0.5833	0.6529	-1.32	-1.7691	0.0441
6	ATT+UMP+KMS	0.0392	8.406	0.5444	0.6154	-1.5139	-1.9661	0.0279
7	UNI+UMP+KMS	0.0451	15.0817	0.4667	0.7112	-0.5774	-1.8081	0.0448
8	UNI+UMP+KMS	0.044	15.0501	0.4444	0.7434	-0.5357	-1.299	0.0704
9	ATT+PCA+KMS	0.0391	6.729	0.6556	0.5372	-3.2638	-3.4085	-0.0228
10	UNI+PCA+KMS	0.0495	19.356	0.4667	0.7117	-0.3987	-2.4635	0.0249

TABLE 5.5: Comparison of finalist hyperparameter setting combinations cont.

decided to implement a qualitative topic merging process seeking to increase the minimum cluster size to five data points and consolidate some of the clusters where appropriate. This enabled us to arrive at a much more consolidated threat taxonomy whilst still retaining a high level of diversity between clusters and coherence within clusters. Ultimately this further improves classification accuracy during feature importance analysis and allows for a stratified 70:30 split between training and test data, improving the validity of its evaluation.

This topic merging process sought to merge clusters containing similar topic keywords and example CWE entries. We developed a formal method which sought to ensure that only objective and appropriate criteria were used to justify a merging of two or more clusters, ensuring it represents a robust and repeatable process. This merging method is as follows:

1. Identify representative CWEs, and their CWE parent classes, for each of the original 57 topics.
2. Give a representative description to each topic based on the representative keywords, the representative documents (the original CWEs) and the corresponding parent CWEs. These were developed to be broad enough to cover the representative documents and keywords, but otherwise as narrow as possible.
3. For each topic, identify all similar topics with reference to the following:
   (a) Clustering proximity using a hierarchical clustering diagram of topics (i.e. whether they were directly adjacent, or had the same parent cluster at a higher level of abstraction)
   (b) Similarity between two topics as represented in the similarity matrix
   (c) Evaluation of the representative CWEs' similarity
   (d) Evaluation of the key terms' similarity
4. For each pairing of similar topics, determine whether to cluster according to one of the following ways:
   (a) Merge a smaller cluster into a pre-existing broader cluster.
   (b) Merge multiple clusters into a new and broader cluster.
5. Continue this process (by looping from step 3) until no further merges are feasible without losing the essential and significant distinguishing characteristics of the topic (threat).
6. For each of the new or amended topics, provide a representative description of each.

During the topic merging process, as described above, the hierarchical clustering diagram, as shown in Figure 5.2, was used in conjunction with the similarity matrix (Figure 5.3, the topic descriptions, keywords and document (i.e. CWE) samples, in order to conduct the cluster merging process. Some of these pieces of key information used in the cluster merging process are provided in Table C.2 of Appendix C. Throughout the cluster merging we observed that the representative CWEs for each topic cluster were often closely related and often had the same parent CWE, which gave an early indication that the clustering through topic modelling had identified meaningfully distinct topics. While the hierarchical clustering diagram proved helpful in identifying potential appropriate cluster merges, we found that it could certainly not be relied upon to identify appropriate merges. For example, the merged threat topic, described as 'Improper adherence to coding best practice' in Table 5.6 was derived from original topic ID 8, however topics from a broad range of top-level clusters were merged into it, indicating that this topic is a cross-cutting topic and was not effectively identified as a unified topic during topic modelling. Other clusters had relatively few samples merged into them, indicating the corresponding topic had been identified well during topic modelling.

By implementing our topic merging method, we were able to reduce the number of topic clusters down to 19 and ensure each cluster had a minimum of five data points. The particular merging decisions made are given in Table C.3 of Appendix C, and the resultant topics are illustrated in Table 5.6. We suggest that these results demonstrate the utility of using NLP, and topic modelling in particular, in cyber security research, including for the automated and repeatable generation of concise cyber threat taxonomies for characterising the evolving threat landscape.

5.4 Discussion

The approach used in this chapter of using an LLM to generate a multi-dimensional dataset based on an unstructured text-based original data source, combined with topic modelling and cluster merging, has performed well for the purposes of characterising a knowledge domain into a concise set of topic clusters which simultaneously retain a broad scope and expressiveness when compared to the original dataset. This leads us to expect that this approach will also be suitable for characterising original datasets of a similar structure, irrespective of their particular knowledge domain.

There is a notable limitation on the repeatability of these results, however, by virtue of the use of the LLM for pre-processing the text-based inputs to the topic modelling process. In particular, different LLM foundation models will perform differently in this respect, producing different outputs. Further, even a single model will produce some degree of variation in successive prompts worded in exactly the same way. This is

Topic	Merged threat description	Count
0	Missing or improper neutralisation	54
2	Improper input validation	18
3	Insecure product configuration	18
4	Insecure memory buffer	14
5	Improper session management	14
6	Weak implementation or use of password-based authentication	20
8	Improper adherence to coding best practice	69
9	Insufficient privilege management or compartmentalisation	14
10	Side channel and SoC threats	14
11	Insufficient or misconfigured access control	12
14	Missing authentication	8
15	Error or feature omission in UI	7
16	Insufficient technical documentation	14
17	Improper adherence to database best practice	7
18	Deficient hardware architecture/security features	32
20	Improper control of software resource through its lifetime	24
21	Incorrect permission assignment for critical resource	7
22	Improperly configured or weak encryption	16
32	Improper or weak credential management	5

TABLE 5.6: The resultant topics after the merging process, including the topic ID of the original topic it is derived from and a count of the number of samples in each merged cluster.

due to the inherent probabilistic nature of the generation of tokens in the transformer-based architecture. As discussed in Section 7, it would be interesting for future work to evaluate these variations, and also to provide validation of the text-based descriptions of the 12 threat attributes using an alternative method or data source as comparator.

Despite the fact that the field of cyber security is a relatively narrow field within which to identify distinct topics, we consider our approach to have yielded satisfactory results. In particular, we were able to characterise the resultant topics using coherent text-based descriptions. BERTopic does provide a set of keywords associated with each topic cluster, however determining coherent human-readable descriptions of the meaning of each cluster has always been a difficult challenge. However, as per the merged threat descriptions in Table 5.6, the final threat classes after our cluster merging process had distinct topics which we were qualitatively able to describe concisely.

5.5 Summary

In this Chapter, we have demonstrated the successful implementation of a rigorous method for evaluating various models and hyperparameter settings for conducting topic modelling of a dataset of cyber weaknesses. Our key findings are as follows:

- The optimal configuration for our final topic model employed the default configuration of BERTopic, namely the use of UMAP for dimensionality reduction and HDBSCAN for clustering, with tuned hyperparameters.
- We were able to perform a cluster merging process to reduce the number of clusters from 57 to 19, while retaining a high level of diversity between clusters and coherence between clusters. This resulted in a minimum of 5 data points per cluster.
- The 19 clusters represented in Table 5.6 are an appropriate concise characterisation of the threat landscape, and the qualitative human-readable descriptions provided by the authors represent their meaning to an appropriate degree of scope and expression.

Chapter 6

Threat Characterisation with Feature Importance Analysis

As set out Chapter 3, RQ3 relates to the challenge of characterising the essential nature of the concept of a cyber threat by quantifying the relative importance of 12 cyber threat attributes. This chapter is partly based on joint work with Tsang et al. with whom we have published some of the main results in Tsang et al. (2024). The threat attributes under consideration are those shown in Table 5.1 of Chapter 5.

In view of this challenge, we have translated the dataset arrived at through the clustering task described in Chapter 5 into a classification paradigm and have harnessed a variety of feature importance techniques in order to quantitatively address this research question. Feature importance analysis techniques are typically used within the context of classification, rather than clustering, and we could not identify any viable feature importance techniques for clustering. Thus, in order to bypass such a limitation, we were required to adopt this classification paradigm.

As a classification task, this research activity involves training a machine learning model to perform classification in order to predict the clustering labels from the topic modelling stage based on their respective attribute-specific BERT embeddings (i.e. their feature values). In order to achieve this, we generated BERT embeddings for each of the 12 attributes of each of the 503 CWE entries and used the clustering labels of each CWE entry, after conducting topic modelling, to train our classifier. This approach is deemed to offer suitable feasibility for conducting feature importance analysis in the context of topic modelling.

Three classifiers were used which each contained their own 'built-in' technique for measuring feature importance. These classification algorithms were Random Forest, XGBoost and Linear SVM. We further integrated either SHAP and permutation importance into each classifier to widen the scope for conducting comparative analysis

of each classifier. We refer to these as 'external methods' of measuring feature importance. Following feature importance analysis, we included a further step of aggregation to summarise the results. The final feature importance scores were also normalised to relative percentages in order to enable each score for each method to be to be compared on equal terms. This process for threat characterisation with feature importance is illustrated in Figure 6.1.

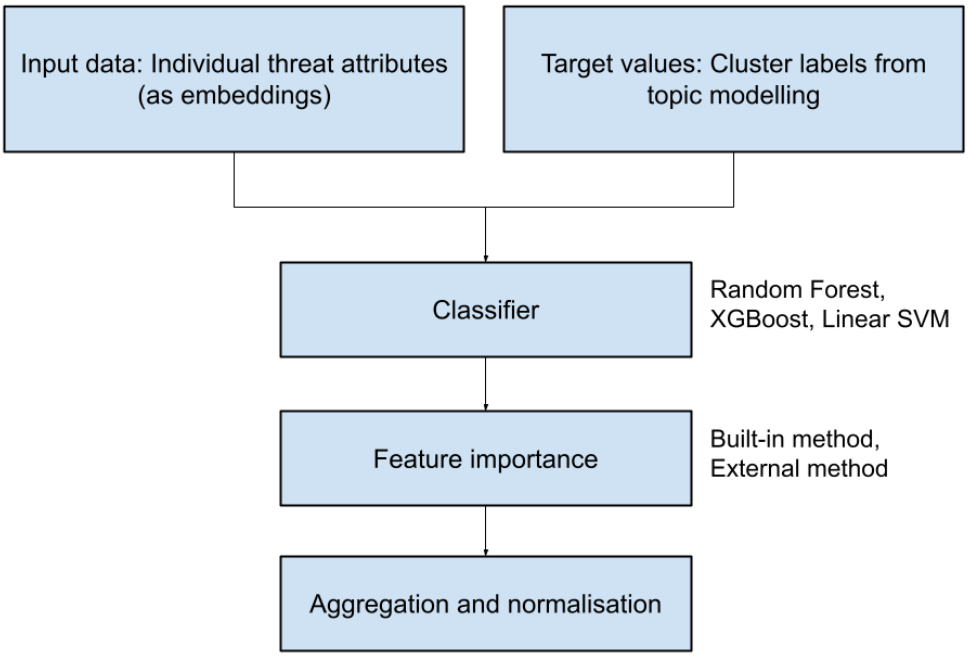


FIGURE 6.1: An illustration of the feature importance analysis process.

6.1 Classifiers for feature importance

As mentioned above, we considered three separate classification models, each of which contains its own built-in method for performing feature importance analysis.

6.1.1 Random Forest

Random Forest is a machine learning algorithm which uses multiple decision trees to perform classification, usually identifying the majority classification prediction from each decision tree in order to determine its ultimate prediction.

Random Forest performs feature importance by evaluating how much each feature contributes to the model's accuracy. This technique measures the mean decrease in Gini 'impurity' across all splits in which a given feature is used. Gini impurity is a measure of how often a random element in the set would be incorrectly labelled at a given node

in the corresponding decision tree. Accordingly, it is a measure of the purity of classification performed in the Random Forest decision tree. In the case of feature importance, if a feature being included within splitting generally has a strong effect at reducing Gini impurity, this will result in a higher feature importance, and vice versa. In Random Forest this is performed over all decision trees to compute an average feature importance for each feature.

6.1.2 eXtreme Gradient Boosting (XGBoost)

Similarly to Random Forest, XGBoost also combines multiple decision trees. However, rather than creating many decision trees in parallel, as with Random Forest, XGBoost creates decision trees sequentially, with each subsequent tree seeking to improve upon the last. Its final classification label is a probability distribution of each class, rather than a majority vote across all decision trees.

While XGBoost comprises built-in mechanisms for feature importance analysis based on coverage, frequency and permutation importance, we focused on its method using 'Gain'. This metric represents how much each feature contributes to improving model accuracy when it is used to split the data at a node in the decision tree. This 'splitting' happens at each node in the decision tree where the model chooses the feature and threshold to 'split' the data in a way that (in the case of classification) maximises the separation between classes.

6.1.3 Linear Support Vector Machine (Linear SVM)

While Support Vector Machines (SVMs) are most commonly used for binary classification, they can also be used for multi-class classification. They work by predicting the hyperplane which most reliably separates samples of different classes. Linear SVM identifies this hyperplane to separate these classes based on a linear combination of features. Therefore, since the weight vector directly indicates how each feature contributes to determining the decision boundary, the weights assigned to each feature can be interpreted as being an indication of how important each feature is to the classification decision.

In the case of multi-class problems, Linear SVM adopts a One-vs-Rest (OvR) strategy in which for k classes, k classifiers are trained wherein each classifier is configured as a binary classifier. For each OvR binary classifier one class is the class being evaluated and the other class is all the rest. When performing classification, the binary classifier with the highest confidence score (i.e. has the highest distance from the hyperplane) is used to make the overall prediction.

In this use-case, the importance of each feature can be calculated for each class individually by looking at the weights from the corresponding binary classifier. Therefore, an average (or sum) of these weights across all binary classifiers can be calculated to determine the global feature importance score for each feature.

Across each of these three classification algorithms, there are a variety of relative strengths and weaknesses. The most notable strength of the built-in methods are their convenience, since the libraries used to train and execute the classifiers also contain the feature importance techniques. A key limitation is that they all have a tendency to misrepresent the importance of highly correlated features due to their inability to differentiate between the contributions of individual features when they contain overlapping information.

6.2 External methods for feature importance

SHapley Additive exPlanations (SHAP) is a unified framework for interpreting machine learning models by assigning each feature an importance value based on its relative contribution to a specific prediction which generally has better performance or conformity with human intuition than previous approaches (Lundberg and Lee (2017)). SHAP is derived from techniques in cooperative game theory and therefore benefits from a rigorous theoretical foundation. Each feature's importance is calculated by averaging its contribution across all possible feature combinations. This makes SHAP more immune from biases arising from the order in which features are introduced. Further, while the built-in feature importance methods described above may not account for interactions between correlated features, SHAP offers a principled way to distribute features, even when features interact.

While SHAP offers several advantages, one limitation is that it has higher requirements for computational resource. We observed that, for the Linear SVM classifier in particuar, SHAPs memory requirements exceeded what was available leading us to consider alternative external methods to evaluate feature importance for Linear SVM.

Permutation importance (Fisher et al. (2018)) is a model-agnostic technique for assessing feature importance by simply evaluating the impact of randomly permuting (shuffling) each feature on the model's performance. Having trained the model and measured its baseline performance, one simply permutes one feature as a time, breaking the relationship between that feature and the target variable, thus disrupting the information in that feature while keeping the rest of the features unchanged. By observing model performance after permutation, it is possible to calculate an importance score for each feature based on the delta between the baseline performance and the post-permutation performance for each feature. This approach is intuitive to understand and inherently accounts for interactions that features might share with each other.

While less efficient than built-in methods, its memory requirements were suitable for adoption as an external method for feature importance using the Linear SVM classifier.

6.3 Training the classifiers

For the purposes of training the classifiers, the input data is the BERT embeddings for each feature of the CWE entries and the clustering labels from the topic modelling stage were used as target variables. Training was performed using a training set with another set reserved as test data. We used 2-fold cross-validation to minimise the risk of overfitting, during which each split was stratified (i.e. the distribution of classes in each split is representative of the original distribution) to ensure each class was adequately represented. This helps to reduce the risk of a training or test set having too few (or zero) examples from a certain class. Due to the low volume of samples in some classes, we could not adopt a stratified *k*-fold (i.e. with 5 or 10 folds) cross-validation, since we need a minimum of 3 samples per class in order for each fold to be sufficiently stratified.

A two-stage process was used to conduct hyperparameter tuning of the various classification models using two libraries from the Scikit-learn (Pedregosa et al. (2011)) package. In the first stage the 'RandomisedSearchCV' library was used to conduct cross-validation using a randomised search of hyperparameter settings for each model. During cross-validation, we identified the randomised hyperparameter values which produced the highest accuracy classifiers during cross-validation. During the second stage, these values were refined by testing values slightly higher and lower than those identified in the first stage to find the optimal settings for each hyperparameter. The specific hyperparameters which each classification model uses are mostly different, however this approach was applicable across all three classifiers.

6.4 Classifier evaluation

In sum, we used a total of six methods to measure feature importance:

1. Random Forest using its built-in method
2. Random Forest with SHAP
3. XGBoost using its build-in method
4. XGBoost with SHAP
5. LinearSVM using its built-in method
6. LinearSVM with permutation importance

	Random Forest	XGBoost	Linear SVM
CV Score	0.65	0.65	0.71
Accuracy	0.62	0.72	0.82

TABLE 6.1: A table showing the cross-validation and accuracy scores for the three classifiers.

In order to evaluate these feature importance scores across all methods, it was necessary to normalise and aggregate them. In particular, in order to compare results from each method, we needed to normalise them to ensure a common scale was used. This comprised summing the importance values across all embeddings relating to a given feature and dividing it by the sum of all importance values across all embeddings for all features. This enabled us to arrive at a percentage value for each attribute for any given method, allowing them to be compared on equal terms.

Having normalised the scales through this process, it was possible to evaluate the relative importance of the 12 features (threat attributes) across each of the six methods. Further, we produced a representation based on the rank of each feature among the other features, rather than its percentage importance, enabling us to more easily identify certain patterns, or differences between methods, that were harder to spot using just their percentage feature importance values.

We also conducted an evaluation of each classifier based on its cross-validation (CV), as determined during training, and its classification accuracy scores, using the reserved test data. As shown in Table 6.1, the results of this evaluation indicate that the Linear SVM model exhibits superior performance, followed by XGBoost and Random Forest respectively.

6.5 Feature importance analysis

Having trained the three classifiers, we conducted feature importance analysis using both a built-in method and an external method for each classifier, yielding a total of six methods for measuring feature importance. Figure 6.2 illustrates the results of this analysis, normalised as relative percentages. It also comprises box plots illustrating the quarterlies of the distributions of relative percentages for each cyber threat attribute (i.e. feature).

As shown in Figure 6.2, based on the results of our approach, the description of the particular *vulnerability* of a given CWE represents the single most important feature for characterising a cyber threat. This is followed by its *technical impact*, affected *security properties*, relevant *countermeasures* and *detection methods*. Although less so, the *method*, *attack vector(s)* and relevant *asset(s)* are also important.

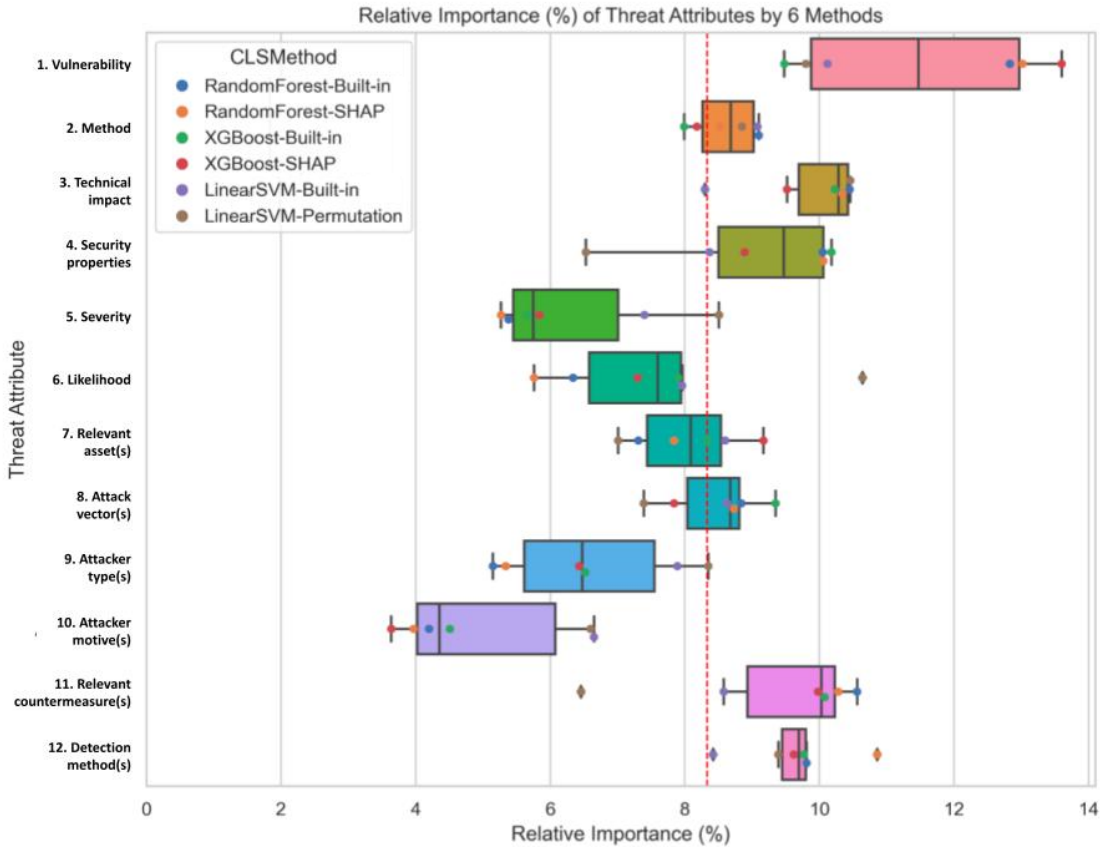


FIGURE 6.2: An illustration of the relative importance of each of the 12 threat attributes across all 6 methods of feature importance analysis.

The red dotted line in this Figure marks the 8.33% point which indicates the point of equal distribution among the 12 threat attributes. Therefore attributes close to or above this point are indicated to be the most important. Accordingly, the threat attributes distinctly beneath this line are helpful when characterising threats in that they do have some importance in classification accuracy, however our results indicate that they are less important than those previously mentioned.

We observed that the range of values for relative importance for some methods is considerably smaller than for others. This can make it harder to evaluate the relative importance of each feature, since methods with a narrower range lead to an underrepresentation of the differences between more or less important attributes. Therefore, another box plot showing the ranks of relative importance for each attribute was produced, as shown in Figure 6.3. In this diagram, lower ranks indicate greater relative importance, with a rank of 6.5 being shown with a red dotted line indicating the median rank.

This importance rank reveals that while many attributes follow the same patterns shown in the relative importance results, there are a number of key outliers. For example, attribute 11 (*relevant countermeasures*) generally shows a high rank, yet for one method (the Linear SVM model using permutation for feature importance) it ranks last. The

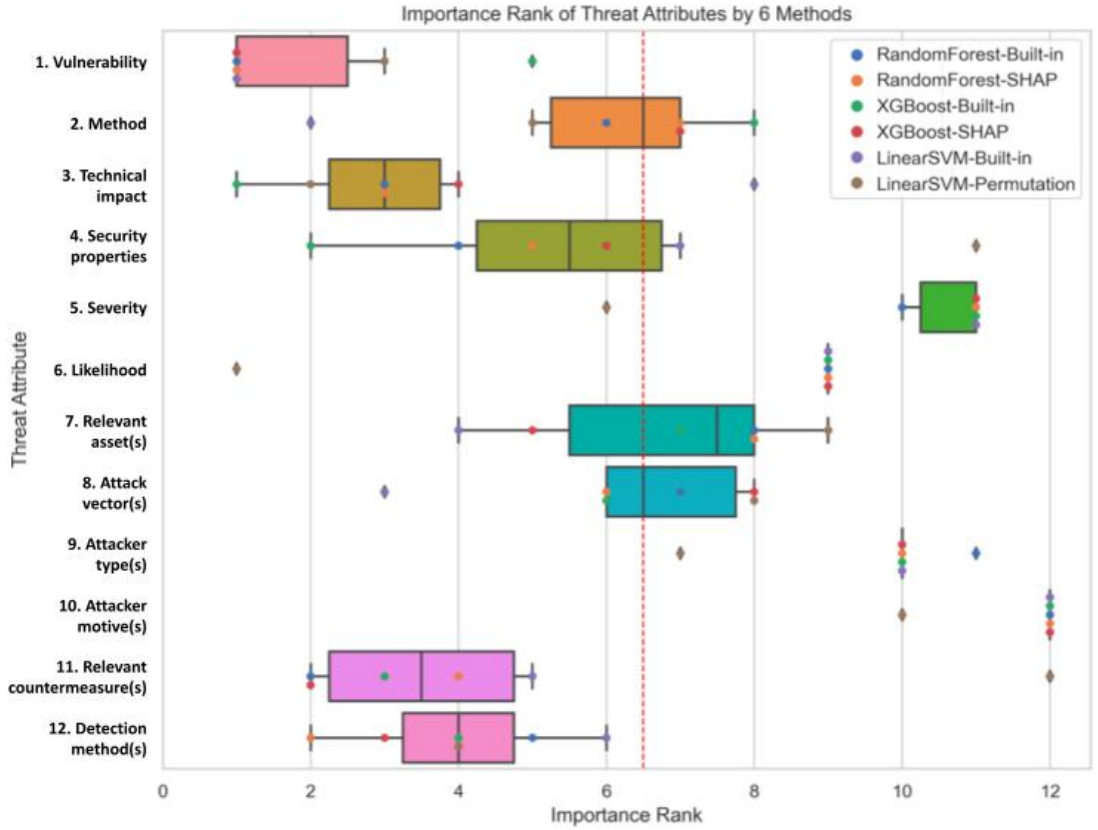


FIGURE 6.3: An illustration of the importance rank of each of the 12 threat attributes across all 6 methods of feature importance analysis.

same method ranks attribute 6 (*likelihood*) 1st despite the other methods ranking it 9th, and it ranks attribute 5 (*severity* 6th despite others ranking it 10th or 11th. It is notable that the ranks of attribute importance using Linear SVM, whether the built-in or permutation methods are used, are distinct outliers when compared to ranks using either of the other models.

A simple average of the relative importance values, as a means of deriving our final results, is appealing in that it incorporates all methods. However, since different classifiers have different scales this would have biased some models over others. Further, based on a qualitative evaluation of the feature importance results, we had a number of doubts about the efficacy of Linear SVM as a classification model.

Despite Linear SVM's high classification accuracy, we were concerned that its narrow scope of relative importance values, which ranged from approximately 6% to 10%, was significantly narrower than the other two models, which were generally within the 4% to 13% range. Such a narrow range was considered to hamper the effective differentiation of attributes. This may be due to the limitations of a linear classifier of being less able to capture complex non-linear interactions between features, which could lead to greater variation in importance values. Further, SVMs seek to identify the hyperplane that maximises the margin between classes and they do so by finding a balance across

all features. This means that it may not capture the distinctiveness of some features as strongly as other types of classification, ultimately leading to smaller variations in feature importance. Therefore, we chose to exclude the Linear SVM results from further evaluation.

In contrast, the recursive splitting inherent to tree-based methods such as Random Forest and XGBoost means that the few features which lead to optimal splits are emphasised. They are generally more sensitive to identifying complex feature interactions and their non-linearity allows them to capture those nuanced characteristics more accurately, leading to more pronounced variation in feature importance. The results of both Random Forest and XGBoost using the SHAP method are shown in Figure 6.4.

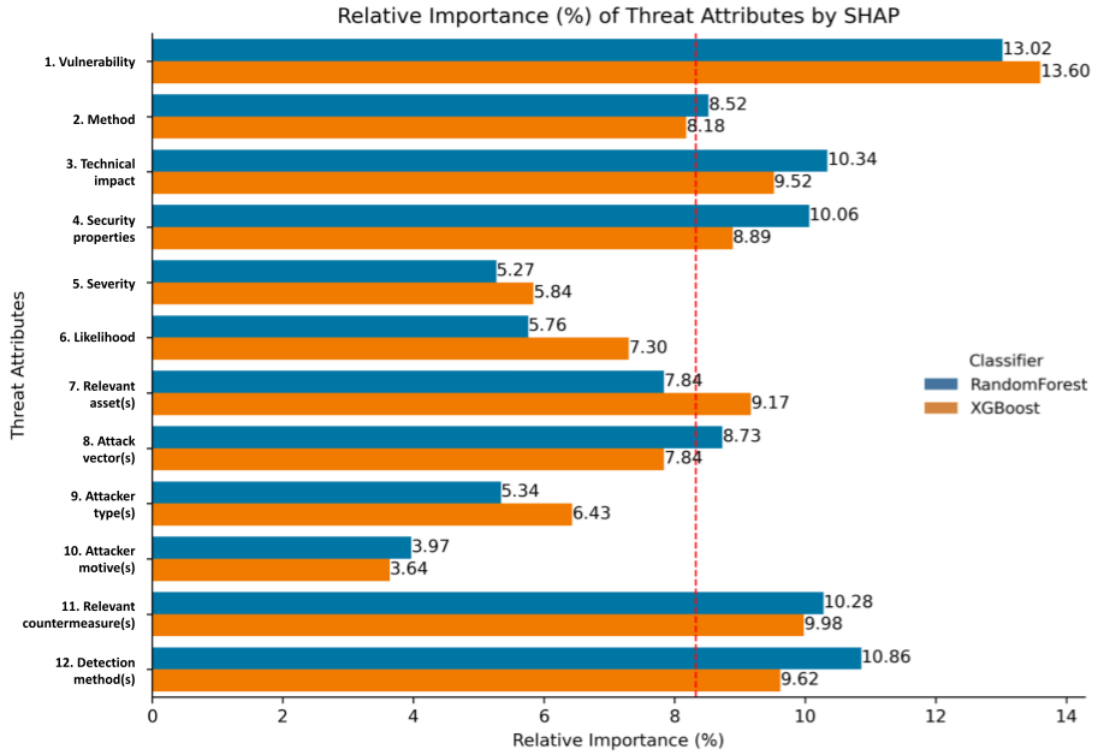


FIGURE 6.4: An illustration of the relative importance of each of the 12 threat attributes with the SHAP method using both the Random Forest and XGBoost classifiers.

We considered using an average of both Random Forest and XGBoost using SHAP for the final feature importance scores, however due to their relatively similar results and the superior classification performance of XGBoost, we selected XGBoost with SHAP as the our method for measuring feature importance. Accordingly, the final rank of all 12 attributes based on their feature importance, along with their relative feature importance scores, are shown in Table 6.2.

Rank	Attribute ID	Feature name	Relative importance
1	1	Vulnerability	13.60
2	11	Relevant countermeasure(s)	9.98
3	12	Detection method(s)	9.62
4	3	Technical impact	9.52
5	7	Relevant assets	9.17
6	4	Security properties	8.89
7	2	Method	8.18
8	8	Attack vector(s)	7.84
9	6	Likelihood	7.30
10	9	Attacker type(s)	6.43
11	5	Severity	5.84
12	10	Attacker motive(s)	3.64

TABLE 6.2: Ranking and relative feature importance for all 12 attributes using XGBoost with SHAP.

6.6 Asset-based 'core' threat model

On the basis of the results of our feature importance analysis process outlined in Section 6.5 we have developed a partial ontology which is a general threat model for use in a broad range of asset-based threat modelling activities. This ontology was developed based on the ontology engineering principles set out in Chapter 2. In the context of ReFAThM (see Chapter 4) this ontology is generalised in that it could be adopted as the 'core model' of any cyber threat model. As such, it may be referred to as an 'ontology' (albeit, without a fuller set of properties and facets), a 'generalised asset-based threat model' or a 'core [threat] model' in the remainder of this Chapter.

Although we do not seek to develop a complete ontology in this chapter - and therefore have not comprehensively used any particular method - we have developed a partial ontology with reference to the principles discussed in Noy and Mcguinness (2001), Fernández-López et al. (1997), Gruber (1995) and Blanco et al. (2011). In respect of the ontology engineering phases of Fernández-López et al. (1997), as described in Section 2.5.2, we are concerned only with the 'Specification' and part of the 'Conceptualisation' phases. In particular, this was done by:

1. Specification
   (a) Identifying the purpose of the ontology.
   (b) Identifying the domain and scope of the ontology.
2. Conceptualisation
   (a) Enumerating the important terms.
   (b) Defining the classes in the ontology, and

(c) Defining the (in our case, limited) hierarchical and non-hierarchical relationships between classes.

In this case, the purpose is to represent the relationship between the key attributes of a cyber threat in a generally extensible manner, to enable automated reasoning, prediction, and improved decision-making. The domain of this ontology is the 'cyber threat'. The scope of the ontology covers the broadest range of threat modelling use cases which the ontology may be extended to serve, including representing the relationship between attacker type and the likelihood of a technical impact on an asset, or the impact on technical impact of a countermeasure being applied to an asset to mitigate a vulnerability. Competency questions may typically be defined here. However, since we are not developing a complete ontology, this was not deemed necessary. The important terms were considered simply to be the 12 threat attributes, all of which were selected to be classes in the ontology.

Typically, defining the relationships between classes in an ontology is one of the most important aspects to complete. Where decisions have been made in relation to non-intuitive hierarchical and non-hierarchical relationships between classes, these have been justified in the next section. The relationships between classes are identified as follows:

1. $\langle Countermeasure \rangle \langle AppliedTo \rangle \langle Asset \rangle$
2. $\langle Countermeasure \rangle \langle Mitigates \rangle \langle Vulnerability \rangle$
3. $\langle Vulnerability \rangle \langle Affects \rangle \langle Asset \rangle$
4. $\langle DetectionMethod \rangle \langle Detects \rangle \langle Vulnerability \rangle$
5. $\langle DetectionMethod \rangle \langle AppliedTo \rangle \langle Asset \rangle$
6. $\langle AttackVector \rangle \langle Targets \rangle \langle Asset \rangle$
7. $\langle Method \rangle \langle Uses \rangle \langle AttackVector \rangle$
8. $\langle AttackerType \rangle \langle Uses \rangle \langle AttackVector \rangle$
9. $\langle AttackerType \rangle \langle Uses \rangle \langle Method \rangle$
10. $\langle AttackerType \rangle \langle Has \rangle \langle AttackMotive \rangle$
11. $\langle Asset \rangle \langle Has \rangle \langle TechnicalImpact \rangle$
12. $\langle TechnicalImpact \rangle \langle Has \rangle \langle Likelihood \rangle$
13. $\langle TechnicalImpact \rangle \langle Affects \rangle \langle Securityproperties \rangle$
14. $\langle TechnicalImpact \rangle \langle Has \rangle \langle Severity \rangle$

Further to the definition of inter-class relationships, the conceptualisation phase would typically require defining the properties, facets and their data types for each class at this stage, as well as appropriate taxonomies for creating class instances. However, since we are not implementing a full ontology, this was not completed. The output of our ontology engineering process is shown in Figure 6.5, which illustrates our ontology representing a generalised core asset-based threat model.

In the illustration of our ontology of Figure 6.5, the dark grey concepts indicate the 6 most important attributes for characterising a threat, while the white concepts illustrate the 6 least important attributes. In the bottom-right of each concept, the relative feature importance of each attribute is given as a percentage, as per the results in Table 6.2. The essential concepts forming the ontology are those identified as possible threat attributes in Table 5.1. The essential relationships between these concepts have been represented and labelled.

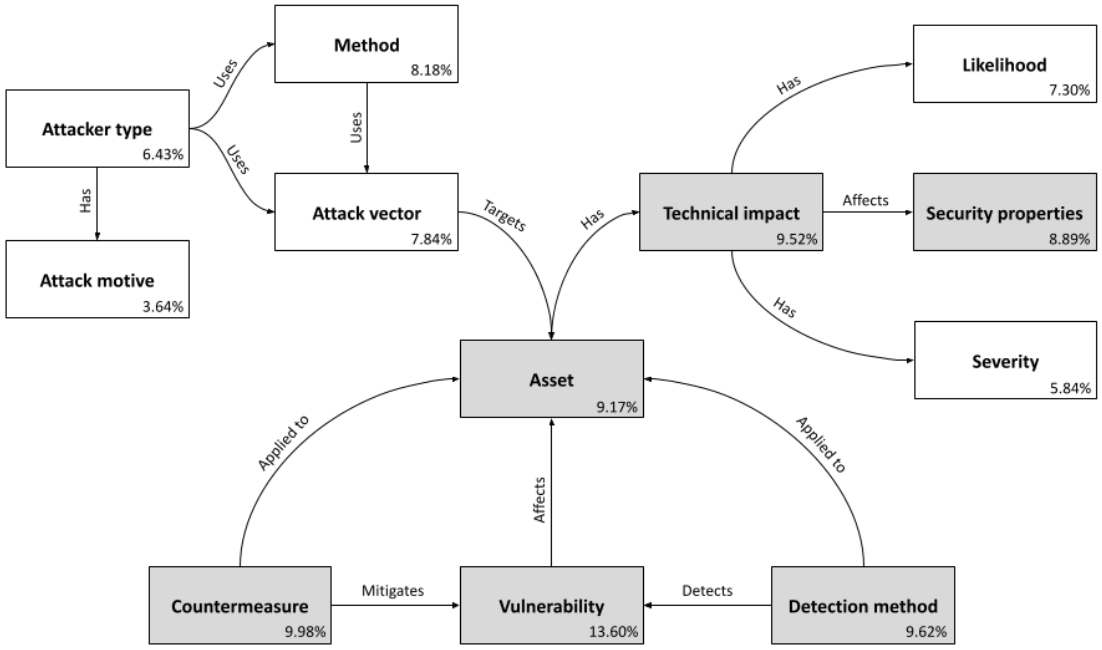


FIGURE 6.5: An ontology representing a generalised asset-based threat model (i.e. 'core threat model') based on the findings of our feature importance analysis.

In addition to the ontology representation of Figure 6.5, a formal ontology specification is provided in OWL using the Turtle syntax. The classes and relationships between classes of the ontology are defined in the OWN file of Listing 6.1. The validity of that syntax and semantic completeness of the core model as represented by this ontology has been demonstrated using WebVOWL (version 1.1.3), of which there is an illustration provided in Figure 6.6. This shows that such a model is suitable for extension with further concepts, properties and other facets, and for the construction of a knowledge base, in a variety of asset-based threat modelling applications.

```
1 @prefix : <http://example.org/ontology#> .
```

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

##############################
# Declaration of the Ontology
##############################
:CoreAssetBasedThreatOntology rdf:type owl:Ontology .

##############################
# Classes
##############################
:Countermeasure     rdf:type owl:Class .
:Asset              rdf:type owl:Class .
:Vulnerability      rdf:type owl:Class .
:DetectionMethod    rdf:type owl:Class .
:AttackVector       rdf:type owl:Class .
:Method             rdf:type owl:Class .
:AttackerType       rdf:type owl:Class .
:AttackMotive       rdf:type owl:Class .
:TechnicalImpact    rdf:type owl:Class .
:SecurityProperties rdf:type owl:Class .

##############################
# Object Properties
##############################

# (1) Countermeasure Applied to Asset
:appliedTo
    rdf:type owl:ObjectProperty ;
    rdfs:domain :Countermeasure ;
    rdfs:range :Asset .

# (2) Countermeasure Mitigates Vulnerability
:mitigates
    rdf:type owl:ObjectProperty ;
    rdfs:domain :Countermeasure ;
    rdfs:range :Vulnerability .

# (3) Vulnerability Affects Asset
:affects
    rdf:type owl:ObjectProperty ;
    rdfs:domain :Vulnerability ;
    rdfs:range :Asset .

# (4) Detection method Detects Vulnerability
:detects
    rdf:type owl:ObjectProperty ;
    rdfs:domain :DetectionMethod ;
```

```
    rdfs:range :Vulnerability .

# (5) Detection method Applied to Asset
:appliedToAsset
    rdf:type owl:ObjectProperty ;
    rdfs:domain :DetectionMethod ;
    rdfs:range :Asset .

# (6) Attack vector Targets Asset
:targets
    rdf:type owl:ObjectProperty ;
    rdfs:domain :AttackVector ;
    rdfs:range :Asset .

# (7) Method Uses Attack vector
:usesAttackVector
    rdf:type owl:ObjectProperty ;
    rdfs:domain :Method ;
    rdfs:range :AttackVector .

# (8) Attacker type Uses Attack vector
:usesAttackVectorAttacker
    rdf:type owl:ObjectProperty ;
    rdfs:domain :AttackerType ;
    rdfs:range :AttackVector .

# (9) Attacker type Uses Method
:usesMethod
    rdf:type owl:ObjectProperty ;
    rdfs:domain :AttackerType ;
    rdfs:range :Method .

# (10) Attacker type Has Attack motive
:hasAttackMotive
    rdf:type owl:ObjectProperty ;
    rdfs:domain :AttackerType ;
    rdfs:range :AttackMotive .

# (11) Asset Has Technical impact
:hasTechnicalImpact
    rdf:type owl:ObjectProperty ;
    rdfs:domain :Asset ;
    rdfs:range :TechnicalImpact .

# (12) Technical impact Affects Security properties
:affectsSecurityProperties
    rdf:type owl:ObjectProperty ;
    rdfs:domain :TechnicalImpact ;
    rdfs:range :SecurityProperties .

# (13) Technical impact Has Likelihood
```

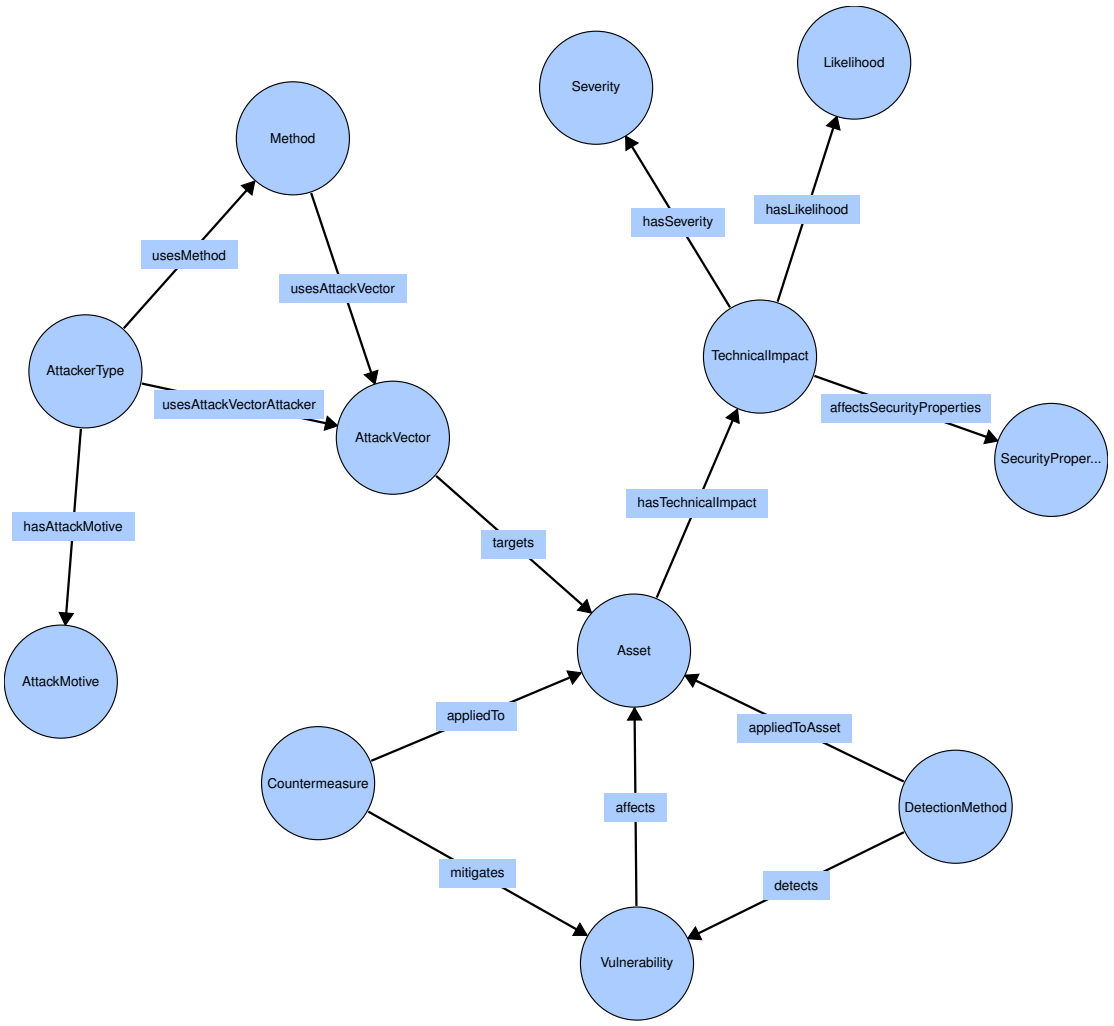


FIGURE 6.6: A graph representing the asset-based threat modelling ontology.

```
:hasLikelihood
    rdf:type owl:ObjectProperty ;
    rdfs:domain :TechnicalImpact ;
    rdfs:range :Likelihood .

# (14) Technical impact Has Severity
:hasSeverity
    rdf:type owl:ObjectProperty ;
    rdfs:domain :TechnicalImpact ;
    rdfs:range :Severity .
```

LISTING 6.1: An OWL file using Turtle syntax representing our asset-based cyber threat ontology

6.7 Evaluation and discussion

We suggest that the results of our feature importance analysis and the resultant ontology offer a hugely valuable and insightful representation of the fundamental characteristics of a cyber threat and have a broad range of use cases in fields right across the domain of cyber security. In this section, we discuss some of the particularly notable observations and implications of the results of this chapter.

Our results indicate that the *vulnerability* related to a cyber threat represents the single most important feature for characterising, and thus discriminating between, cyber threats. Based on our methodology, this means that the description of the *vulnerability* plays the most important role in classifying a CWE entry into a particular threat class. This matches our prior expectations, since according to common definitions, cyber threats are enabled by the existence of a vulnerability (or type of vulnerability, i.e. a weakness) in a system. Further, as might be expected of an asset-based threat model, the *asset* concept is right at the centre of our 'core model' and the other attributes considered in this study have relatively obvious and necessary relations with respect to the *asset* concept.

In general, the features shown to be the most important which are part of our core threat model are those which intuitively form the most essential concepts, in terms of their proximity to the concept of a *vulnerability* and its manifestation as a threat in the context of a particular asset (as per the approach taken with asset-based threat modelling). While many of the ontologies discussed in Chapter 2 include the concepts of *countermeasure*, *vulnerability* and *asset* in their core models, the quantification of feature (i.e. ontology concept) importance as represented in our ontology reveal that *detection method* is also an incredibly powerful feature for characterising cyber threats. Our study does not discuss this topic in depth, or propose any particular ways in which a detection method can be represented in a more complete threat model. However this does indicate that this concept would be a key priority when developing more complete, effective and appropriate asset-based threat models in the future.

The countermeasure which is applied to an asset to mitigate a vulnerability also plays a considerable role in characterising a threat, despite the threat existing independently of the countermeasure. This may be due to the fact that the suitability of a given countermeasure is contingent on the specific vulnerability and asset which the threat relates to, both of which are important threat attributes themselves.

As a collection of properties, the *type*, *motive*, *method* and *vector* used by the attacker appear to be of less relevance compared to the concepts which relate more closely to how a vulnerability is manifested within the asset to produce a technical impact. This is likely to be due to the fact that a broad range of attackers could represent a very similar threat insofar as they target the same asset by exploiting the same vulnerability.

The *method* and *attack vector* are concepts in our ontology which are closely related and have very similar levels of feature importance. This may be due to the relative ambiguity of the definition of a *method* which could have lead GPT-3.5 to generate similar feature descriptions for each of them. Likewise, the concepts of *technical impact* and *security properties* are also closely related and share similar relative importance scores. This is may be due to their similar meaning. We consider *security properties* to be a subset of the *technical impact* of a threat, albeit an important one.

While a given threat may generally have a similar *likelihood* and *severity* (i.e. risk) there is also a considerable range in both these attributes for a given threat. For example, a threat (as such) appears to be less contingent upon the particular attacker type and motive than upon the technical aspects relating to the threat's emergence. However, the likelihood of a threat is highly contingent upon the attacker type and motive, since that would impact factors such as how well resourced they are and whether they will seek to realise the threat. As such, closely related threats may have considerably different likelihoods. Likewise, a variety of threats may share very similar severities, rendering severity a less important feature for meaningfully differentiating between threats. This likely explains why these attributes have a relatively low importance score.

With reference to ReFAThM (4.6), we can observe that our ontology is consistent with the definition of a 'core model', and allows the specific composition (i.e. 'asset(s)') and behaviour (i.e. 'technical impact') of a threat, as well as that of controls, to be represented in terms of their relationship to the assets (i.e. the 'IT system characterisation') which a threat model is concerned with. As such, our core model could form the basis of an asset-based cyber threat model by representing such threats in the form of 'security patterns' (Fernandez-Buglioni (2013)) using the ontology of our core model as its pattern template.

The methodological approach used in this chapter comprises the use of a variety of classifiers and hyperparameter settings, which combined offer an appropriate degree of optimisation and validation their collective results. However, since the original CWE dataset was pre-processed using a large language model, resulting the generated dataset, our feature selection (including the dominant feature - the sample's 'vulnerability') is therefore also reliant on the analysis of this generated data. While we have taken steps to validate our feature importance results using multiple classifiers, there is an inherent limitation in this approach. Our suggestions for further verifying our results to measure this uncertainty is set out in Chapter 7.

6.8 Summary

In this Chapter, we have demonstrated the successful implementation of a robust method for evaluating various six methods for conducting feature importance analysis of a data

set of cyber weaknesses. Our key findings are as follows:

- In spite of the high dimensionality of our sample training set, our three classifiers were satisfactorily accurate, each falling with the range of 60-80%.
- While the accuracy of the Linear SVM classifier was higher than both Random Forest and XGBoost, Linear SVM has a narrower range of feature importance scores and resulted in more outliers than the other models. This indicates that Linear SVM is less able to sufficiently represent and model the interaction between features.
- The feature importance method SHAP demonstrated similar values for relative feature importance when using both the Random Forest and XGBoost classifiers. Therefore, SHAP was selected as the final feature importance method, with the respective built-in methods serving as baselines for comparative analysis. Further, since XGBoost exhibited superior classification accuracy to Random Forest, we selected XGBoost as our final classifier.
- The five most important features in accurately classifying each CWE into the correct threat category (as determined in Chapter 5) were the *vulnerability*, *relevant countermeasure(s)*, *detection method(s)*, *technical impact* and *relevant asset(s)*.
- These results are generally consistent with the authors' expectations, and are qualitatively justifiable in terms of their mutual significance in forming the basis of a 'core threat model' as illustrated in Figure 6.5.

Chapter 7

Conclusions

This chapter summarises the key results which address our research questions. It also discusses our key contributions and makes recommendations for future work, both by improving the existing methodological approach, and by using our results to inform the exploration of new research avenues.

7.1 Conclusions

Asset-based cyber threat modelling is a prevalent and mature activity in both academia and industry, and offers distinct advantages for the purposes of characterising IT systems and modelling the impacts of cyber threats on those systems. While various threat modelling frameworks and ontology engineering principles have been published, our review of the background literature identified that, to our knowledge, there was no framework for asset-based threat modelling for assisting in the development and evaluation of asset-based threat models.

Further, while there are a broad range of comprehensive sources of cyber threat intelligence, these data sources are extremely large, rendering them difficult to represent in a structured manner for the purposes of conducting automated asset-based threat modelling. There is also little consensus about the proper manner to integrate such knowledge bases into a core 'threat model', whether that be an ontology or other form of semantic model.

Lastly, our literature review identified that there are a broad range of asset-based threat models which exist, including several 'core models'. However, they were generally produced using a qualitative process which is hard to reproduce and validate. In particular, there was no robust technique for quantifying the relative importance of threat attributes for the purpose of characterising cyber threats. While there are several quantitative techniques for evaluating risks to provide decision support, existing techniques

do not seek to measure which attributes of a threat are the most important to consider when approaching asset-based threat modelling. In view of this, we identified the research questions described in Section 3.1 as being of considerable importance to furthering the field of asset-based cyber threat modelling.

The research activities undertaken have involved approaching key questions in asset-based threat modelling from a novel perspective. This has included applying methods in machine learning and natural language processing to answer questions relating to the semantic structure and content of the cyber threat landscape. Within the scope of our work, this has been a challenging field to address with regards to verifying the validity of our results, since each of our three research questions are open to a degree subjective qualification and are also broad in their scope. However, we have developed an overall approach, presented in this thesis, which comprises complementary methods to answer these related research questions, and which each adopts feasible approach to verification as described in previous chapters.

In particular, we used a systematic literature review to validate the reference framework of Chapter 4 giving rise to a quantified comparison of successive rounds of the SLR process, leading to minor refinements of ReFAThM during that process. Further, the research question of Chapter 5 addresses a challenging gap in the literature relating to characterising the threat landscape, and adopt a comparative evaluation of a wide range of topic models and hyperparameter configurations, validating that the key findings are broadly reflected across the relevant comparators. Further, the results of Chapter 6 based on XGBoost were likewise validated based on a comparative evaluation against a RandomForest classifier, and multiple feature important techniques. The final 'core model' ontology of that chapter is examined in terms of its relation to the components of ReFAThM, further validating the coverage and expressiveness of that reference framework. While the core model is only a partial ontology, this is fully consistent with the iterative nature of ontology engineer, and enables further detail (facets and properties) to be added based on the requirements of particular use cases. Thus, based on its congruity with both ReFAThM and other core ontologies in the literature, our core model has been appropriately validated. Sections 7.3 and 7.4 of this chapter set out the limitations in our methodological approach and the future work that could provide additional validation of its key results.

The following points summarise our conclusions in relation to these research activities addressing those questions.

- The key general activities which are necessary for performing asset-based threat modelling were found, in Chapter 4, to include: *defining the goal*, *defining the key terms of the model*, *extracting a core model*, *identifying threats* (including both their *behaviour* and *composition*), *characterising the IT system* being modelled, *identifying*

the controls being modelled (including their *behaviour* and *target*), and finally performing a *validation* activity.

- In Chapter 5, we demonstrated that a multi-attribute text-based threat data source can be characterised through clustering using topic modelling and cluster merging, in order to generate a concise cyber threat knowledge base which retains a high degree of breadth and expressivity in relation to the original data set, and to do so in a highly repeatable manner.

- This topic modelling activity found that the CWE data set can be consolidated into 19 classes representing primary threat types represented in that data set. This process can be repeated regularly, as the cyber threat landscape continues to evolve, in order to provide researchers with an up-to-date cyber threat knowledge base which can be applied to research challenges within, and external to, the domain of threat modelling.

- The 19 threats identified include threat types such as *improper input validation*, *insecure product configuration*, *insecure memory buffer*, *weak implementation or use of password-based authentication*, *insufficient privilege management or compartmentalisation*, *improper adherence to database best practice* and *deficient hardware architecture/security features*. These are defined at a high level of abstraction, enabling them to be represented in a generalised manner within a relatively simple asset-based threat model.

- We have also demonstrated, in Chapter 6, that feature importance analysis can be used to quantify the relative importance of cyber threat attributes, and that this informs the development of a 'core threat model'. This approach is shown to be appropriate and helpful, since its findings are consistent with, and therefore further validate, the key components of asset-based threat modelling which constitute ReFAThM.

- Based on our findings, XGBoost appears to offer the most suitable classifier for performing feature importance analysis, with Random Forest also being suitable. Linear SVM is unlikely to offer comparable classification accuracies for conducting feature importance analysis of multi-attribute text-based datasets.

- Based on our feature importance analysis, the 6 most important attributes for constructing a core cyber threat model include its *vulnerability*, *relevant countermeasure(s)*, *detection method(s)*, *technical impact*, *relevant asset(s)* and *security properties*. The model shown in Figure 6.5 represents a 'core threat model' which meets the requirements of the corresponding component of ReFAThM, and can therefore constitute the basis of a core model of a wide range of asset-based threat models in future research.

These findings contribute considerable new information to the field of asset-based cyber threat modelling which will benefit the accuracy and validity of cyber threat modelling activities and outcomes. They also demonstrate the value of further research combining techniques from machine learning and natural language process to the field of cyber threat modelling. Further To the best of our knowledge, our methodology combining multi-attribute data synthesis, topic modelling and feature importance is unique and distinctly generalisable for characterising knowledge domains using unstructured text-based datasets in a range of domains, including those outside of the field of cyber security.

7.2 Contributions

This research has made the following contributions which are beneficial to the research community:

1. A review and discussion of the background literature discussing various approaches to asset-based cyber threat modelling, and an identification of some key research gaps.
2. A review and discussion of the principles, methods and evaluation criteria related to ontology engineering in the context of asset-based threat modelling.
3. A novel Reference Framework for Asset-based Threat Modelling (ReFAThM), to guide the development and evaluation of asset-based cyber threat models.
4. A novel method for quantifying the characterisation of a knowledge domain, such as that of cyber security, involving the synthesis of a multi-attribute dataset, topic modelling and cluster merging, followed by classification and feature importance analysis.
5. A concise, broad and expressive taxonomy of cyber threat types, for use within threat modelling and cyber threat intelligence for academic and commercial applications.
6. A quantification of the relative importance of 12 cyber threat attributes to assist in addressing the challenge of threat characterisation.
7. A generalised asset-based 'core' cyber threat model, to form the basis of context-specific threat models in further research or commercial use cases.

7.3 Limitations and improvements

This research was begun during the early weeks following the release of GPT-3.5, and before many alternative LLMs were available for public use. In view of advancements in LLMs since that stage, a possible extension to this work would be to identify and evaluate a range of alternative LLMs (such as the latest frontier models from top-tier ML labs). Further, alternative BERT-based models for topic modelling, or dimensionality reduction techniques, could be evaluated to identify opportunities to optimise that aspect of our approach. The high degree of modularity of our approach is well suited to facilitating this.

Regarding the dataset generation using GPT-3.5, although we subjectively assess GPT-3.5 to offer plausible descriptions for each attribute, we have not validated the accuracy of its descriptions in the scope of this study. We further observed that GPT 3.5 struggles to offer specificity with regards to the descriptions many of these threat attributes. For example, the descriptions of 'security properties' for almost all CWEs mentions the risks to all the 'CIA' security properties (confidentiality, availability and integrity). Further, descriptions of 'severity' and 'likelihood' generally stated that they depend on various factors which GPT did not know about. This suggests that there is greater work to do to develop more refined prompt engineering to generate our multi-attribute threat descriptions and/or to augment this LLM-based approach with a subject-matter expert validation process to produce a hybrid dataset. This, for example, could involve defining each of the 12 threat attributes more specifically in the context window in order to provide a more expressive and consistent interpretation of those terms for each CWE sample. Alternatively, the relative performance improvements of using model fine-tuning or retrieval augmented generation (RAG) could be investigated.

Another avenue for improvement relates to widening the range of aspects of the BERTopic library which are varied and evaluated, in order to seek to optimise its performance. For instance, BERTopic includes built-in features for performing outlier reduction and cluster merging would could be explored further.

The topics which BERTopic identifies can be fine-tuned in a variety of ways. However, LLMs could also be used to fine-tune these topics, in particular, by generating labels and summaries of each topic. This could be done using prompt engineering and could be an alternative method for assisting in generating the final threat taxonomy and/or for assisting with - or validating - topic merging.

Further work should involve validation of the topic merging process. While the process used to conduct topic merging has been defined explicitly and is relatively objective to implement, we have only performed this process once and have not validated its

accuracy. Such a validation process could be conducted by asking independent subject-matter experts to conduct the same process and measuring the variance in the number, scope and descriptive labels of the final clusters.

7.4 Future work

The research described here offers a range of novel contributions to the field of asset-based threat modelling and can form the basis of future research in this field. Some of those research directions to consider are discussed below.

- In view of the pace of advancements in frontier LLMs and other NLP-based techniques, a clear avenue for future work would be to implement some of the improvements to the current method described in the previous section of this chapter. This could lead to the optimisation of each stage of our pipeline, ensuring the final results of a threat type taxonomy and relative importance scores of each threat attribute are as accurate as possible.
- Although ReFAThM has been validated and refined based on the findings of a systematic literature review, further work could be undertaken to use ReFAThM in the context of a selection of case studies in order to qualitatively evaluate its utility and effectiveness. This could further involve a comparison of how ReFAThM performs by comparison to other asset-based threat modelling approaches (i.e. those which are not based on a reference framework).
- Further, a complete threat modelling process could be defined and evaluated using ReFAThM as its basis. This could include providing guidance for IT system characterisation, threat and controls identification, and characterising the behaviour of those threats and controls. It could then be validated using a case study based on a real threat modelling scenario.
- The data pipeline used in this work was mostly implemented in Python using JupyterLab. The academic field would benefit from this tooling being ported to a different platform and implemented as a command-line tool, as well as hosted in the form of a web-application. This would enable researchers to re-implement our process as new CWE entries come out, ensuring the threat type taxonomy is always up-to-date. Further, if the tooling itself was generalised, a web application could be used to enable third parties to characterise their own datasets in a domain-agnostic manner, in order to perform quantitative characterisation of that domain.
- Although this work considered 12 possible threat attributes, and has identified the 6 attributes which are the most important, we suggest that there could be

many other attributes outside the range of the 12 considered which are relevant for characterising threats. Therefore, future work could involve identifying additional data sources and collating other potential threat attributes from a broader range of sources. A suitable approach for this would be to conduct a systematic literature review to identify further threat attributes. This could help to identify new threat attributes not currently widely accepted as being important for asset-based threat modelling.

- Given the theoretically generalisable nature of our quantitative characterisation method, further work could explore the suitability and effectiveness of using it to characterise adjacent domains, to create taxonomies and even feature importance in fields such as computational biology, law, medicine, cognitive science and the social sciences. In these kinds of fields, our technique could be used to characterise critical knowledge domains, generating broad, expressive and concise taxonomies and ontologies, to aid both academic research and professional practice in these fields.

- Further, even in the field of threat modelling, the topic modelling approach described herein could be applied to characterising the CVE dataset, rather than the CWE dataset. While CVEs represent specific instances of vulnerabilities, rather than more generalised weaknesses, the key benefit is that new CVEs are added to the CVE dataset every week, meaning that this process could be run more frequently, leading to the resultant threat type taxonomy reflecting an even more up-to-date evolving threat landscape. This would also enable the comparative analysis of our approach when using different original data sources.

- Based on the partial ontology ('core threat model') developed in Chapter 6, it would be of interest to evaluate the extent to which our proposed approach differs from other asset-based threat modelling approachs. In particular, it would be a valuable source of validation to determine the extent to which existing asset-based threat models share similarity with ours, and to conduct an evaluation of the reasons for any divergent features (i.e. concepts or inter-class relationships).

- The 'core threat model' identified in Chapter 6 has a robust foundation in our novel method. However, further work could consider how this model could be augmented to produce a complete ontology. This research would then implement that ontology in an appropriate manner, such as using semantic web technologies described in Chapter 2, in order to evaluate its suitability for performing automated reasoning over a cyber threat knowledge base. This would involve specifying properties for each class and identifying relevant taxonomies, followed by implementing each class in a formal specification (such as by using OWL or RDF with a tool such as Protégé). Based on the taxonomies and properties used for each class, external sources of threat intelligence (such as CVE, CVSS, STIX,

CAPEC, CPE and so on) may be used. The ultimate goal would be to develop an automated reasoning tool, using such a knowledge base, for evaluating the relative effectiveness of various countermeasures at mitigating known threats. We anticipate that frontier LLMs could be used to accelerate the integration of relevant taxonomies and construct a comprehensive knowledge base in support of this research activity.

- A further avenue for future research that could either build upon the previous suggestion, or be implemented independently from it, would be to explore the potential for graph neural networks (GNNs) and various ML-based inference engines to be used to conduct automated threat prediction and evaluation over that knowledge base.

- Much of the original motivation for this research arose from the need to develop tooling to evaluate the effectiveness of the controls defined in the NCSC's Cyber Essentials (CE) scheme. Accordingly, an immediately relevant piece of further work relates to the development of a complete ontology for countermeasure evaluation which is suitable for encoding the meaning of the CE controls into a suitable model. This project could use ReFAThM as the basis of an ontology enineering process for asset-based threat modelling; extending the core model of Chapter 6) using the ontology engineering methodological findings of Chapter 2 to construct a complete ontology for this purpose, and using the threat classes determined in Chapter 5 to construct a knowledge base consistent with that ontology in order to generate a complete semantic model for evaluating the effectiveness of the CE controls.

Appendix A

Systematic Literature Review Results

This Appendix comprises an additional Figure relevant to the systematic literature review described in Chapter 4.

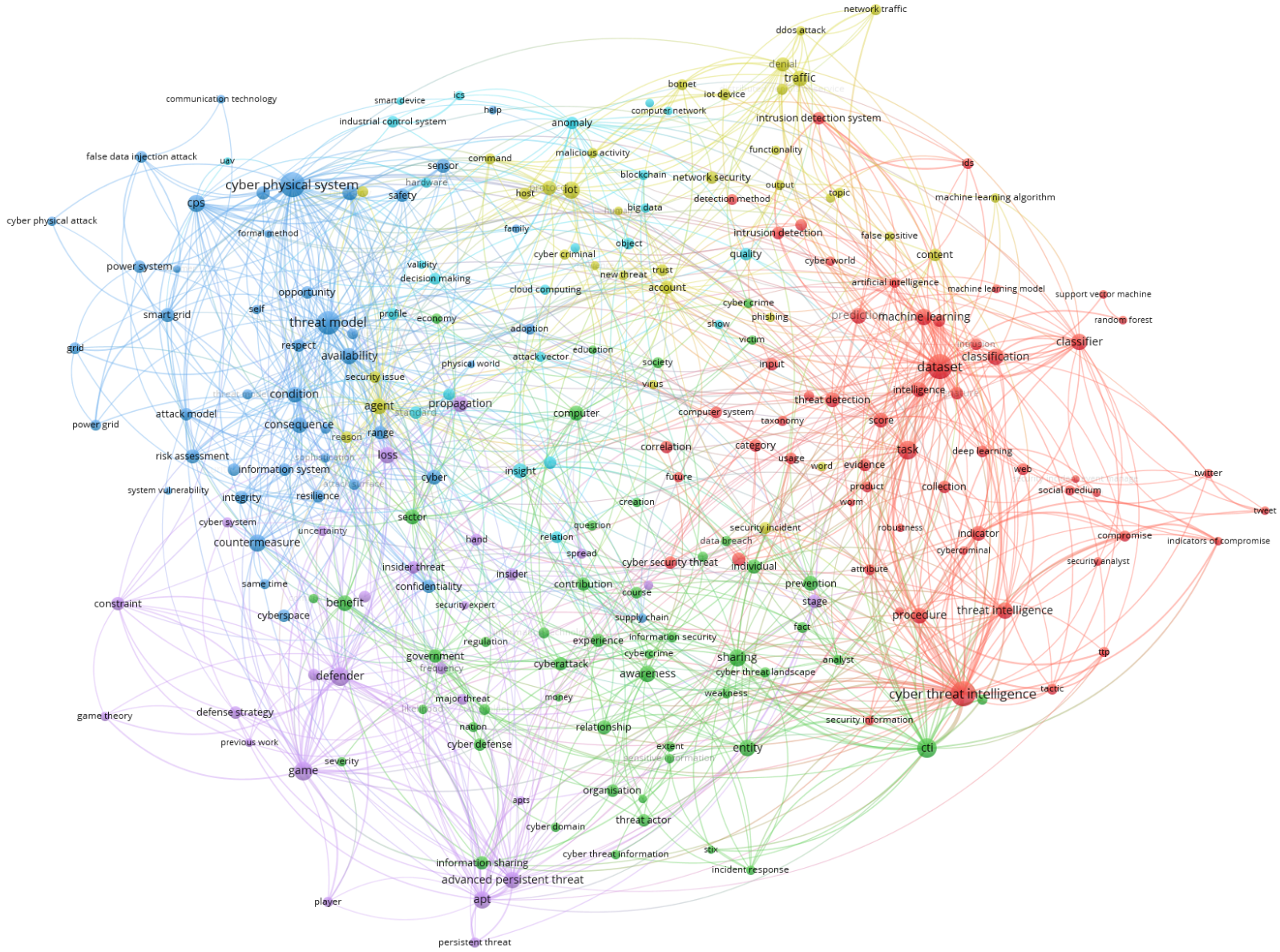


FIGURE A.1: An enlarged illustration of the the keyword analysis and clustering performed on papers relating to threat modelling in the Microsoft Academic Graph publications database, as visualised using VOSViewer. A more concise version is provided in Figure 4.1 of Chapter 4

.

Appendix B

Systematic Literature Review Configuration

This Appendix comprises additional Figures and details the configuration of each of the four searches conducted as part of the systematic literature review described in Chapter 4.

IEEE Xplore	
Description	This database covers electrical engineering, computer science and electronics, and indexes more than 160 journals and 1200 conference proceedings.
Search tool	The 'Command Search' feature
Link	https://ieeexplore.ieee.org/search/advanced/command
Search fields	'metadata'
Document type	Journal article
Query	("All Metadata": "threat modelling" OR "All Metadata": "threat modeling" OR "All Metadata": "threat model")
Results returned	95
Date of search	14 March 2022

TABLE B.1: Configuration of IEEE Xplore search

Scopus	
Description	This database claims to deliver 'the broadest coverage of any interdisciplinary abstract and citation database'. It covers 240 disciplines and indexes 20,000 peer-reviewed journals and 5.5 million conference papers.
Search tool	The 'Advanced Search' feature
Link	https://www.scopus.com/search/form.uri?display=advanced
Search fields	'TITLE-ABS-KEY'
Document type	Journal article
Query	TITLE-ABS-KEY ({threat model}) OR TITLE-ABS-KEY ({threat modeling}) OR TITLE-ABS-KEY ({threat modelling}) AND (LIMIT-TO (PUBSTAGE , "final")) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SRCTYPE , "j")) AND (LIMIT-TO (EXACTKEYWORD , "Threat Modeling") OR LIMIT-TO (EXACTKEYWORD , "Threat Model")) & OR & TITLE-ABS-KEY ({threat modeling}) & OR & TITLE-ABS-KEY ({threat modelling}) & AND & (LIMIT-TO (PUBSTAGE , & "final")) & AND & (LIMIT-TO (DOCTYPE , & "ar")) & AND & (LIMIT-TO (SUBJAREA , & "COMP")) & AND & (LIMIT-TO (LANGUAGE , & "English")) & AND & (LIMIT-TO (SRCTYPE , & "j")) & AND & (LIMIT-TO (EXACTKEYWORD , & "Threat Modeling") & OR & LIMIT-TO (EXACTKEYWORD , & "Threat Model"))
Results returned	149
Date of search	14 March 2022

TABLE B.2: Configuration of Scopus search

Springer Link	
Description	This database indexes over 8.3 million scientific documents. It contains journals and conference proceedings published by the Springer publishing house.
Search tool	The 'Advanced Search' feature
Link	https://link.springer.com/advanced-search
Search fields	Manually search within the title, abstracts and keywords fields
Document type	Journal article
Query	"threat modeling" OR "threat model" OR "threat modelling"
Results returned	24
Date of search	14 March 2022

TABLE B.3: Configuration of Springer Link search

Web of Science	
Description	This database covers over 12,000 high-impact journals and over 150,000 conference proceedings.
Search tool	The 'Advanced Search' feature
Link	https://www.webofscience.com/wos/woscc/advanced-search
Search fields	Title, abstract, author keywords, and 'Keywords Plus' which includes close synonyms.
Document type	Journal article
Query	TS=("threat model" OR "threat modeling" OR "threat modelling")
Results returned	200
Date of search	14 March 2022

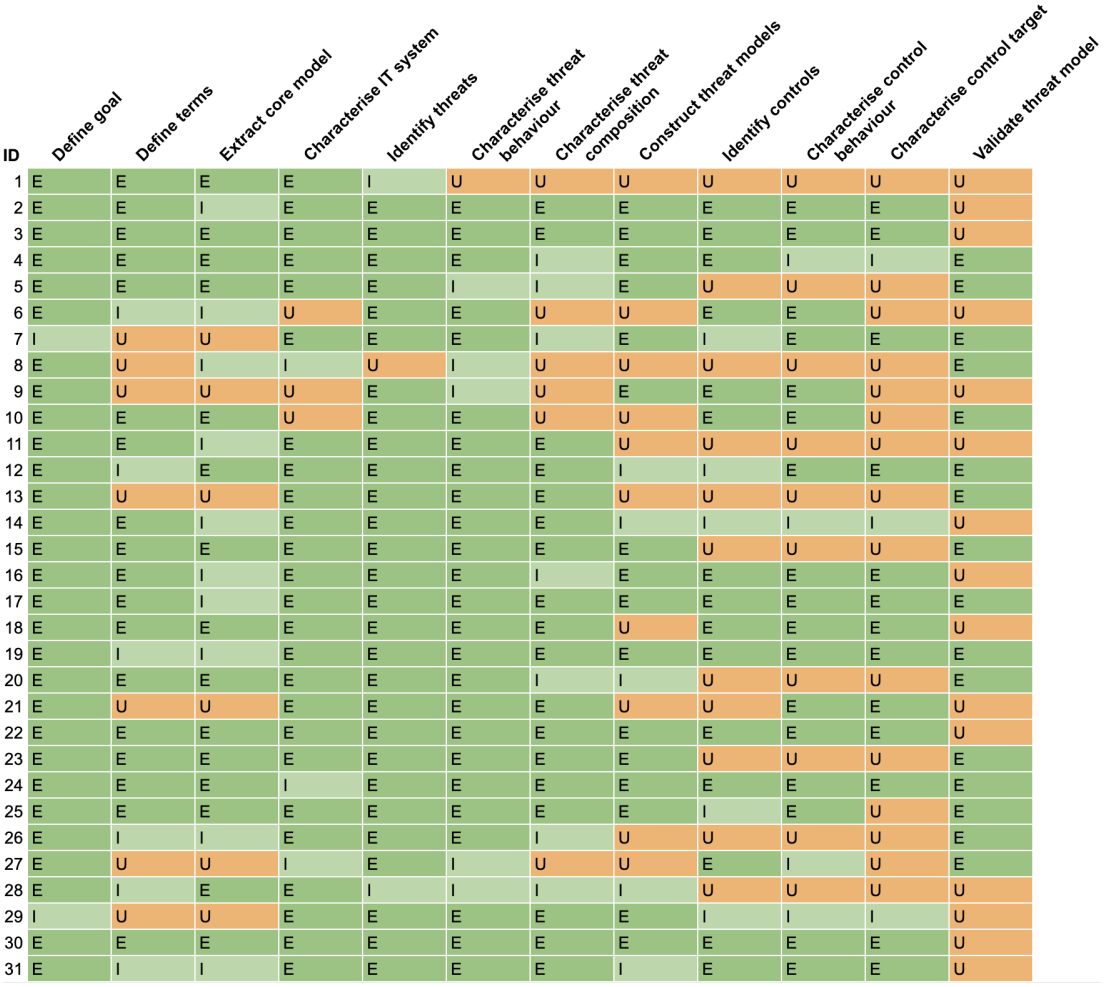
TABLE B.4: Configuration of Web of Science search

ID	Identifier	Define goal	Extract core model	Characterise IT system	Identify threats	Characterise threat	Identify controls	Characterise mitigation	Validate model
1	Zegzhda2015	E	E	E	I	U	U	U	U
2	Rizvi2020a	E	I	E	E	E	E	E	U
3	Jacomme2021	E	E	E	E	E	E	E	U
4	Sciarretta2020	E	E	E	E	E	E	I	E
5	Lai2019	E	E	E	E	I	U	U	E
6	Mayrhofer2021	E	I	U	E	E	E	E	U
7	Xiong2018	I	U	E	E	E	I	E	E
8	Pietro2006	E	I	I	U	I	U	U	E
9	Pei2004	E	U	U	E	I	E	E	U
10	Meszaros2017	E	E	U	E	E	E	E	E
11	Butts2005	E	I	E	E	E	U	U	U
12	Liu2020	E	E	E	E	E	I	E	E
13	Asvija2021	E	U	E	E	E	U	U	E
14	Suleiman2015	E	I	E	E	E	I	I	U
15	Pan2017	E	E	E	E	E	U	U	E
16	Navas2019	E	I	E	E	E	E	E	U
17	Rizvi2020	E	I	E	E	E	E	E	E
18	Li2014	E	E	E	E	E	E	E	U
19	Casola2019	E	I	E	E	E	E	E	E
20	Stellios2021	E	E	E	E	E	U	U	E
21	Torr2005	E	U	E	E	E	U	E	U
22	Hofmann2011	E	E	E	E	E	E	E	U
23	Kamm_ller2017	E	E	E	E	E	U	U	E
24	Xu2006	E	E	I	E	E	E	E	E
25	Joshi2021	E	E	E	E	E	I	I	E
26	Andel2008	E	I	E	E	E	U	U	E
27	Park2010	E	U	I	E	I	E	I	E
28	AlFedaghi2011	E	E	E	I	I	U	U	U
29	Dhillon2011	I	U	E	E	E	I	I	U
30	Rhee2013	E	E	E	E	E	E	E	U
31	Taib2013	E	I	E	E	E	E	E	U

FIGURE B.1: A table illustrating the SLR analysis conducted of the draft framework of Chapter 4 (1 of 2).

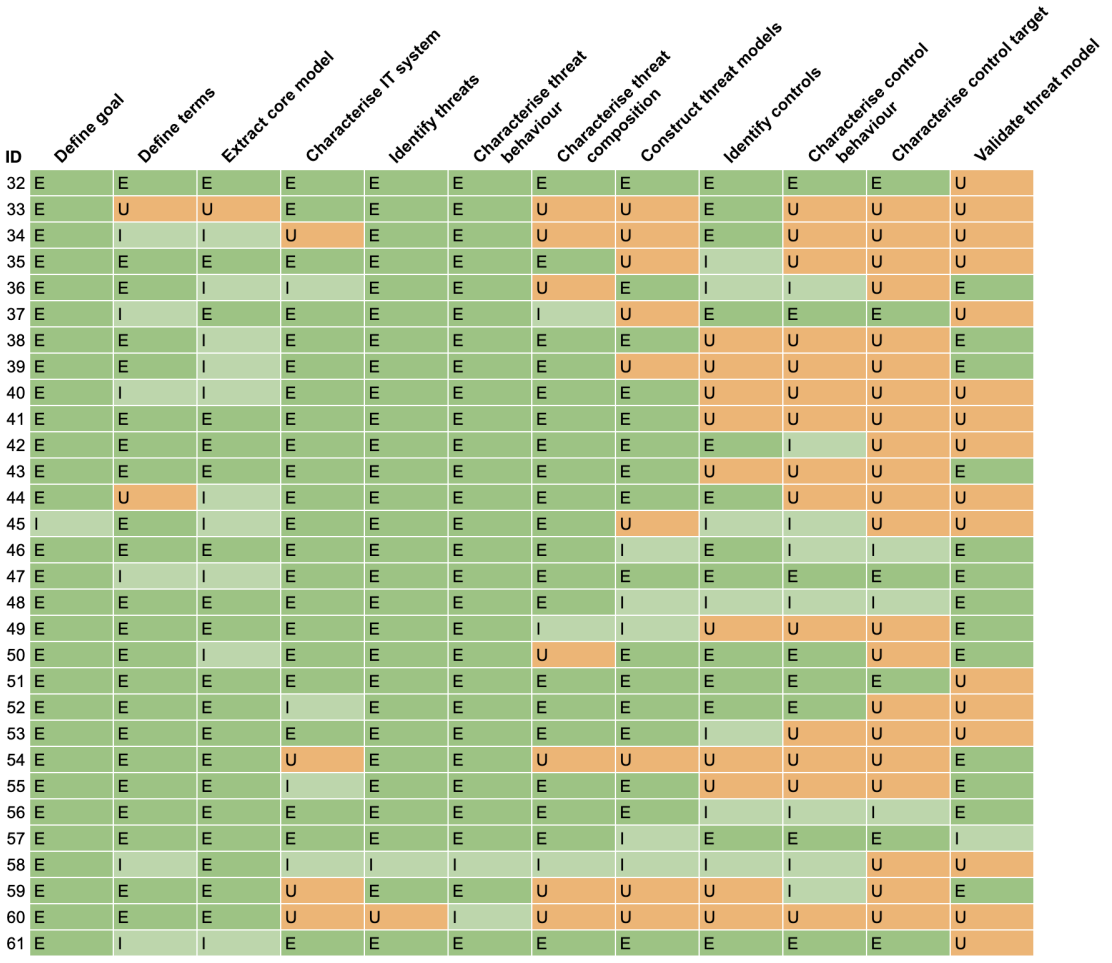
ID	Identifier	Define goal	Extract core model	Characterise IT system	Identify threats	Characterise threat	Identify controls	Characterise mitigation	Validate model
32	Caceres2013	E	E	E	E	E	E	E	U
33	Dimitriadis2013	E	U	E	E	E	E	U	U
34	Pendergrass2014	E	I	U	E	E	E	U	U
35	Sheila2015	E	E	E	E	E	I	U	U
36	Martina2015	E	I	I	E	E	I	I	E
37	Pastrana2015	E	E	E	E	E	E	E	U
38	Anjaria2017	E	I	E	E	E	U	U	E
39	Venkatasen2018	E	I	E	E	E	U	U	E
40	Novokhrestov2019	E	I	E	E	E	U	U	U
41	Hamad2020	E	E	E	E	E	U	U	U
42	Joshi2020	E	E	E	E	E	E	I	U
43	Egoshin2020	E	E	E	E	E	U	U	E
44	Sharma2021	E	I	E	E	E	E	U	U
45	Viswanathan2021	I	I	E	E	E	I	I	U
46	Husnoo2021	E	E	E	E	E	E	I	E
47	Xiong2022	E	I	E	E	E	E	E	E
48	Jbair2022	E	E	E	E	E	I	I	E
49	Rak2022	E	E	E	E	E	U	U	E
50	Potteiger2016	E	I	E	E	E	E	A	E
51	Saitta2005	E	E	E	E	E	E	E	U
52	Elahi2009	E	E	I	E	E	E	E	U
53	Wang2009	E	E	E	E	E	I	U	U
54	Razzaq2014	E	E	U	E	E	U	U	E
55	Undercoffer2003	E	E	I	E	E	U	U	E
56	Surridge2013	E	E	E	E	E	I	I	E
57	Herzog2007	E	E	E	E	E	E	E	I
58	Gyrard2013	E	E	I	I	I	I	I	U
59	Gao2013	E	E	U	E	E	U	I	E
60	Bromander2016	E	E	U	U	I	U	U	U
61	Caralli2007	E	I	E	E	E	E	E	U

FIGURE B.2: A table illustrating the SLR analysis conducted of the draft framework of Chapter 4 (2 of 2).



ID	Define goal	Define terms	Extract core model	Characterise IT system	Identify threats	Characterise threat behaviour	Characterise threat composition	Construct threat models	Identify controls	Characterise control behaviour	Characterise control target	Validate threat model
1	E	E	E	E	I	U	U	U	U	U	U	U
2	E	E	I	E	E	E	E	E	E	E	E	U
3	E	E	E	E	E	E	E	E	E	E	E	U
4	E	E	E	E	E	E	I	E	E	I	I	E
5	E	E	E	E	E	I	I	E	U	U	U	E
6	E	I	I	U	E	E	U	U	E	E	U	U
7	I	U	U	E	E	E	I	E	I	E	E	E
8	E	U	I	I	U	I	U	U	U	U	U	E
9	E	U	U	U	E	I	U	E	E	E	U	U
10	E	E	E	U	E	E	U	U	E	E	U	E
11	E	E	I	E	E	E	E	U	U	U	U	U
12	E	I	E	E	E	E	E	I	I	E	E	E
13	E	U	U	E	E	E	E	U	U	U	U	E
14	E	E	I	E	E	E	E	I	I	I	I	U
15	E	E	E	E	E	E	E	E	U	U	U	E
16	E	E	I	E	E	E	I	E	E	E	E	U
17	E	E	I	E	E	E	E	E	E	E	E	E
18	E	E	E	E	E	E	E	U	E	E	E	U
19	E	I	I	E	E	E	E	E	E	E	E	E
20	E	E	E	E	E	E	I	I	U	U	U	E
21	E	U	U	E	E	E	E	U	U	E	E	U
22	E	E	E	E	E	E	E	E	E	E	E	U
23	E	E	E	E	E	E	E	E	U	U	U	E
24	E	E	E	I	E	E	E	E	E	E	E	E
25	E	E	E	E	E	E	E	E	I	E	U	E
26	E	I	I	E	E	E	I	U	U	U	U	E
27	E	U	U	I	E	I	U	U	E	I	U	E
28	E	I	E	E	I	I	I	I	U	U	U	U
29	I	U	U	E	E	E	E	E	I	I	I	U
30	E	E	E	E	E	E	E	E	E	E	E	U
31	E	I	I	E	E	E	E	I	E	E	E	U

FIGURE B.3: A table illustrating the SLR analysis conducted of the refined framework of Chapter 4 (1 of 2).



ID	Define goal	Define terms	Extract core model	Characterise IT system	Identify threats	Characterise threat behaviour	Characterise threat composition	Construct threat models	Identify controls	Characterise control behaviour	Characterise control target	Validate threat model
32	E	E	E	E	E	E	E	E	E	E	E	U
33	E	U	U	E	E	E	U	U	E	U	U	U
34	E	I	I	U	E	E	U	U	E	U	U	U
35	E	E	E	E	E	E	E	U	I	U	U	U
36	E	E	I	I	E	E	U	E	I	I	U	E
37	E	I	E	E	E	E	I	U	E	E	E	U
38	E	E	I	E	E	E	E	E	U	U	U	E
39	E	E	I	E	E	E	E	U	U	U	U	E
40	E	I	I	E	E	E	E	E	U	U	U	U
41	E	E	E	E	E	E	E	E	U	U	U	U
42	E	E	E	E	E	E	E	E	E	I	U	U
43	E	E	E	E	E	E	E	E	U	U	U	E
44	E	U	I	E	E	E	E	E	E	U	U	U
45	I	E	I	E	E	E	E	U	I	I	U	U
46	E	E	E	E	E	E	E	I	E	I	I	E
47	E	I	I	E	E	E	E	E	E	E	E	E
48	E	E	E	E	E	E	E	I	I	I	I	E
49	E	E	E	E	E	E	I	I	U	U	U	E
50	E	E	I	E	E	E	U	E	E	E	U	E
51	E	E	E	E	E	E	E	E	E	E	E	U
52	E	E	E	I	E	E	E	E	E	E	U	U
53	E	E	E	E	E	E	E	E	I	U	U	U
54	E	E	E	U	E	E	U	U	U	U	U	E
55	E	E	E	I	E	E	E	E	U	U	U	E
56	E	E	E	E	E	E	E	E	I	I	I	E
57	E	E	E	E	E	E	E	I	E	E	E	I
58	E	I	E	I	I	I	I	I	I	I	U	U
59	E	E	E	U	E	E	U	U	U	I	U	E
60	E	E	E	U	U	I	U	U	U	U	U	U
61	E	I	I	E	E	E	E	E	E	E	E	U

FIGURE B.4: A table illustrating the SLR analysis conducted of the refined framework of Chapter 4 (2 of 2).

ID	Identifier	Source	Title	Author(s)	Year
1	Zegzhda2015	Search	Approach to the construction of the generalized functional-semantic cyber security model	Zegzhda, P. D.; Zegzhda, D. P.; Stepanova, T. V.	2015
2	Rizvi2020a	Search	Identifying the attack surface for IoT network	Rizvi, Syed; Orr, R. J.; Cox, Austin; Ashokkumar, Prithvee; Rizvi, Mohammad R.	2020
3	Jacomme2021	Search	An Extensive Formal Analysis of Multi-factor Authentication Protocols	Jacomme, Charlie; Kremer, Steve	2021
4	Sciarretta2020	Search	Formal Analysis of Mobile Multi-Factor Authentication with Single Sign-On Login	Sciarretta, Giada; Carbone, Roberto; Ranise, Silvio; Vigano, Luca	2020
5	Lai2019	Search	Symmetric keyring encryption scheme for biometric cryptosystem	Lai, Yen-Lung; Hwang, Jung Yeon; Jin, Zhe; Kim, Soohyong; Cho, Sangrae; Teoh, Andrew Beng Jin	2019
6	Mayrhofer2021	Search	The Android Platform Security Model	Mayrhofer, Rene; Stoep, Jeffrey Vander; Brubaker, Chad; Kralevich, Nick	2021
7	Xiong2018	Search	False Sequential Command Attack of Large-Scale Cyber-Physical Systems	Xiong, Yinqiao; Yang, Ziyu; Wang, Baoyao; Xun, Peng; Deng, Tiantian	2018
8	Pietro2006	Search	Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks	Pietro, Roberto Di; Mancini, Luigi V.; Mei, Alessandro	2006
9	Pei2004	Search	A framework for resilient internet routing protocols	Pei, D.; Zhang, L. X.; Massey, D.	2004
10	Meszaros2017	Search	Introducing OSSF: A framework for online service cybersecurity risk management	Meszaros, Jan; Buchalcevova, Alena	2017

11	Butts2005	Search	Developing an insider threat model using functional decomposition	Butts, J. W.; Mills, R. F.; Baldwin, R. O.	2005
12	Liu2020	Search	Protection of Sensitive Data in Industrial Internet Based on Three-Layer Local/Fog/Cloud Storage	Liu, Jing; Yuan, Changbo; Lai, Yingxu; Qin, Hua	2020
13	Asvija2021	Search	Security Threat Modelling With Bayesian Networks and Sensitivity Analysis for IAAS Virtualization Stack	Asvija, B.; Eswari, R.; Bijoy, M. B.	2021
14	Suleiman2015	Search	Integrated smart grid systems security threat model	Suleiman, Husam; Alqassem, Israa; Diabat, Ali; Edin, Davor Arnautovic; Svetinovic	2015
15	Pan2017	Search	PMCAP: A Threat Model of Process Memory Data on the Windows Operating System	Pan, Jiaye; Zhuang, Yi	2017
16	Navas2019	Search	Understanding and mitigating OpenID Connect threats	Navas, Jorge; Beltran, Marta	2019
17	Rizvi2020	Search	Threat model for securing internet of things (IoT) network at device-level	Rizvi, Syed; Pipetti, Ryan; McIntyre, Nicholas; Jonathan, Iyonna Todd; Williams	2020
18	Li2014	Search	Unified threat model for analyzing and evaluating software threats	Li, XiaoHong; He, Ke; Feng, Zhiyong; Xu, Guangquan	2014
19	Casola2019	Search	Toward the automation of threat modeling and risk assessment in IoT systems	Casola, Valentina; Benedictis, Alessandra De; Massimiliano, Umberto Rak; Villano	2019
20	Stellios2021	Search	Assessing IoT enabled cyber-physical attack paths against critical systems	Stellios, Ioannis; Kotzanikolaou, Panayiotis; Grigoriadis, Christos	2021
21	Torr2005	Search	Demystifying the threat-modeling process	Torr, P.	2005

22	Hofmann2011	Search	Towards a security architecture for IP-based optical transmission systems	Hofmann, Stefan; Kasseckert, Rudolf	2011
23	Kammller2017	Search	Modeling and Verification of Insider Threats Using Logical Analysis	Kammller, Florian; Probst, Christian W.	2017
24	Xu2006	Search	Threat-driven modeling and verification of secure software using aspect-oriented Petri nets	Xu, D.; Nygard, K. E.	2006
25	Joshi2021	Search	Insider Threat Modeling: An Adversarial Risk Analysis Approach	Joshi, Chaitanya; Aliaga, Jesus Rios; Insua, David Rios	2021
26	Andel2008	Search	Adaptive Threat Modeling for Secure Ad Hoc Routing Protocols	Andel, T. R.; Yasinsac, A.	2008
27	Park2010	Search	Optimal decision-making of countermeasures by estimating their expected utilities	Park, S. R.; Noh, S.	2010
28	AlFedaghi2011	Search	On security development Lifecycle: Conceptual description of vulnerabilities, risks, and threats	Al-Fedaghi, S.; Alkandari, A.	2011
29	Dhillon2011	Search	Developer-driven threat modeling: Lessons learned in the trenches	Dhillon, D.	2011
30	Rhee2013	Search	Threat modeling of a mobile device management system for secure smart work	Rhee, K.; Won, D.; Jang, S.-W.; Chae, S.; Park, S.	2013
31	Taib2013	Search	ICMPV6 vulnerability: The importance of threat model and SF-ICMP6	Taib, A. H. M.; Ali, W. N. A. W.; Shaari, N. S.	2013
32	Caceres2013	Search	A threat model for security specification in security evaluation by ISO/IEC 19791	Caceres, G. H. R.; Teshigawara, Y.	2013
33	Dimitriadis2013	Search	Security for mobile operators in practice	Dimitriadis, C. K.	2013

34	Pendergrass2014	Search	A threat table based assessment of information security in telemedicine	Pendergrass, J. C.; Heart, K.; Ranganathan, C.; Venkatakrishnan, V. N.	2014
35	Sheila2015	Search	Dimension of mobile security model: Mobile user security threats and awareness	Sheila, M.; Faizal, M. A.; Shahrin, S.	2015
36	Martina2015	Search	An adaptive threat model for security ceremonies	Martina, J. E.; dos Santos, E.; Carlos, M. C.; Price, G.; Custudio, R. F.	2015
37	Pastrana2015	Search	DEFIDNET: A framework for optimal allocation of cyberdefenses in Intrusion Detection Networks	Pastrana, S.; Tapiador, J. E.; Orfila, A.; Peris-Lopez, P.	2015
38	Anjaria2017	Search	Quantitative analysis of information leakage in service-oriented architecture-based Web services	Anjaria, K.; Mishra, A.	2017
39	Venkatasen2018	Search	A risk-centric defensive architecture for threat modelling in e-government application	Venkatasen, M.; Mani, P.	2018
40	Novokhrestov2019	Search	Model of threats to computer network software	Novokhrestov, A.; Konev, A.; Shelupanov, A.	2019
41	Hamad2020	Search	SAVTA: A hybrid vehicular threat model: Overview and case study	Hamad, M.; Prevelakis, V.	2020
42	Joshi2020	Search	A comprehensive security analysis of match-in-database fingerprint biometric system	Joshi, M.; Mazumdar, B.; Dey, S.	2020
43	Egoshin2020	Search	A model of threats to the confidentiality of information processed in cyberspace based on the information flows model	Egoshin, N. S.; Konev, A. A.; Shelupanov, A. A.	2020
44	Sharma2021	Search	Security Enhancement in Software Defined Networking (SDN): A Threat Model	Sharma, P. K.; Tyagi, S. S.	2021

45	Viswanathan2021	Search	A hybrid threat model for system-centric and attack-centric for effective security design in SDLC	Viswanathan, G.; Prabhu, P. J.	2021
46	Husnoo2021	Search	Do not get fooled: Defense against the one-pixel attack to protect IoT-enabled Deep Learning systems	Husnoo, M. A.; Anwar, A.	2021
47	Xiong2022	Search	Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix	Xiong, W.; Legrand, E.; Oberg, O.; Lagerstrom, R.	2022
48	Jbair2022	Search	Threat modelling for industrial cyber physical systems in the era of smart manufacturing	Jbair, M.; Ahmad, B.; Maple, C.; Harrison, R.	2022
49	Rak2022	Search	ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems	Rak, M.; Salzillo, G.; Granata, D.	2022
50	Potteiger2016	Known	Software and attack centric integrated threat modeling for quantitative risk assessment	Potteiger, Bradley; Martins, Goncalo; Koutsoukos, Xenofon	2016
51	Saitta2005	Known	Trike v.1 Methodology Document [Draft]	Saitta, Paul; Larcom, Brenda; Eddington, Michael	2005
52	Elahi2009	Known	A Modeling Ontology for Integrating Vulnerabilities into Security Requirements Conceptual Foundations	Elahi, Golnaz; Yu, Eric; Zannone, Nicola	2009
53	Wang2009	Known	OVM: An ontology for vulnerability management	Wang, Ju An; Guo, Minzhe	2009
54	Razzaq2014	Known	Ontology for attack detection: An intelligent approach to web application security	Razzaq, Abdul; Anwar, Zahid; Ahmad, H. Farooq; Latif, Khalid; Munir, Faisal	2014
55	Undercoffer2003	Known	Modeling computer attacks: An ontology for intrusion detection	Undercoffer, Jeffrey; Joshi, Anupam; Pinkston, John	2003
56	Surridge2013	Known	Run-time risk management in adaptive ICT systems	Surridge, Mike; Nasser, Bassem; Chen, Xiayou; Chakravarthy, Ajay; Melas, Panos	2013

57	Herzog2007	Known	An Ontology of Information Security	Herzog, Almut; Shahmehri, Nahid; Duma, Claudiu	2007
58	Gyrard2013	Known	The STAC (security toolbox: Attacks & countermeasures) ontology	Gyrard, Amelie; Bonnet, Christian; Boudaoud, Karima	2013
59	Gao2013	Known	Ontology-based model of network and computer attacks for security assessment	Gao, Jian Bo; Zhang, Bao Wen; Chen, Xiao Hua; Luo, Zheng	2013
60	Bromander2016	Known	Semantic Cyberthreat Modelling	Bromander, Siri; Jsang, A.; Eian, Martin	2016
61	Caralli2007	Known	Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process	Caralli, Richard A.; Stevens, James F.; Young, Lisa R.; Wilson, William R.	2007

TABLE B.5: Final collection of journal articles used in the SLR process

Appendix C

Topic Merging Process

This Appendix comprises tables which relate to the topic representation and cluster merging process as described in Chapter 5.

ID	Rep. CWEs	Count	Keyword representation
-1	1298, 641, 843	136	['access', 'data', 'include', 'unauthorized', 'code', 'unauthorized access', 'product', 'sensitive', 'integrity', 'information', 'attackers', 'potential', 'depends', 'affected', 'availability', 'specific', 'behavior', 'gain', 'proper', 'vary']
0	125, 140, 76	35	['input', 'include', 'access', 'sensitive', 'data', 'information', 'product', 'affected', 'user', 'lead', 'used', 'high lead', 'confidentiality integrity', 'monitoring', 'high', 'include monitoring', 'malicious', 'include input', 'input validation', 'occurring']
1	790, 78, 74	19	['product', 'case', 'input', 'products', 'inputs', 'component', 'data', 'special', 'include', 'special elements', 'unexpected', 'list', 'systems', 'elements', 'monitoring', 'securityrelevant', 'attackers', 'operations', 'depends specific', 'upstream']
2	1287, 1173, 625	15	['input', 'validation', 'regular expression', 'regular', 'expression', 'input validation', 'regular expressions', 'glyphs', 'expressions', 'cryptographic', 'cryptographic operations', 'el', 'encoding mechanism', 'include', 'potential', 'application', 'encoding', 'malicious', 'attackers', 'el statement']
3	1125, 203, 1269	14	['product', 'products', 'include', 'behavior', 'surface', 'sensitive', 'information', 'processed', 'used', 'access', 'affected', 'depends', 'monitoring', 'attackers', 'sensitive information', 'access product', 'handling', 'unexpected', 'potential', 'potentially']
4	824, 466, 788	14	['pointer', 'memory', 'buffer', 'null', 'null pointer', 'code', 'input', 'pointers', 'crashes', 'arbitrary code', 'corruption', 'potential', 'arbitrary', 'memory corruption', 'memory locations', 'valid', 'dereference', 'bounds', 'analysis', 'pointer dereference']
5	488, 613, 425	10	['session', 'web', 'authorization', 'web application', 'server', 'url', 'application', 'session management', 'activex', 'web server', 'unauthorized', 'browser', 'restricted', 'urls', 'users', 'access', 'custom url', 'cookies', 'url scheme', 'resources']

6	549, 309, 916	10	['password', 'passwords', 'authentication', 'login', 'accounts', 'attackers', 'user accounts', 'unauthorized access', 'hashing', 'user', 'systems', 'unauthorized', 'access', 'sensitive', 'login attempts', 'attempts', 'easily', 'users credentials', 'information', 'gain']
7	521, 640, 620	10	['authentication', 'password', 'accounts', 'user accounts', 'user', 'account', 'mechanism', 'authentication mechanism', 'access', 'unauthorized', 'case', 'configuration file', 'lockout', 'account lockout', 'gain access', 'unauthorized access', 'information', 'gain', 'sensitive', 'attempts']
8	394, 440, 393	9	['return', 'error', 'handler', 'software', 'return values', 'feature api', 'behavior', 'unexpected', 'api function', 'feature', 'function', 'testing', 'values', 'incorrect', 'specific', 'incorrect return', 'unexpected return', 'exploiting', 'false', 'api']
9	270, 267, 653	9	['privilege', 'privileges', 'resources functionality', 'access', 'threat', 'resources', 'actions', 'unauthorized', 'privilege escalation', 'escalation', 'functionality', 'levels', 'permissions', 'privilege levels', 'unauthorized access', 'include', 'access control', 'restricted', 'rights', 'control']
10	1300, 1192, 1319	9	['soc', 'device', 'electromagnetic', 'information', 'physical', 'threat', 'confidential', 'debug', 'confidential information', 'access', 'case', 'identifiers', 'vendors', 'sensitive information', 'sensitive', 'internal', 'actions', 'electromagnetic fault', 'fault', 'debug agents']
11	804, 1220	9	['captcha', 'challenge', 'key', 'access', 'cache', 'captcha challenge', 'case', 'unauthorized', 'incorrect tokens', 'tokens', 'unauthorized access', 'restricted areas', 'areas', 'transaction', 'securitysensitive', 'identifier', 'control', 'access control', 'information', 'implementing']
12	544, 396, 397	8	['error', 'handling', 'exceptions', 'error handling', 'code', 'handling code', 'conditions', 'exception', 'complex error', 'exception handling', 'overly broad', 'unhandled', 'broad exceptions', 'broad', 'complex', 'default case', 'overly', 'inconsistent', 'software', 'unintended code']
13	1122, 1124, 1121	8	['code', 'complexity', 'excessive', 'commentedout', 'commentedout code', 'bugs', 'quality', 'complex', 'introducing', 'complex code', 'software', 'codebase', 'undetected', 'harder', 'introduced', 'development', 'issues', 'maintain', 'practices', 'code analysis']

14	322, 325	8	['encryption', 'algorithm', 'communication', 'cryptographic algorithm', 'cryptographic', 'weaker', 'data', 'certificate', 'weaker encryption', 'certificates', 'values', 'product', 'data communication', 'authenticity', 'step', 'key exchange', 'exchange', 'previous values', 'previous', 'party']
15	356, 448, 449	7	['user', 'ui', 'interface', 'user interface', 'warning', 'actions', 'users', 'dangerous', 'dangerous function', 'function', 'ui function', 'action', 'obsolete ui', 'obsolete', 'feedback', 'api', 'api interface', 'falsely', 'feature', 'monitoring user']
16	1118, 1110, 1111	7	['documentation', 'design', 'systemdependent', 'programs', 'systemdependent functionality', 'comprehensive', 'design documentation', 'lack', 'productspecific programs', 'productspecific', 'behavior', 'error', 'inputs', 'error handling', 'product', 'inputs outputs', 'documented', 'outputs', 'products design', 'callable']
17	1072, 1049, 1089	7	['performance', 'database', 'queries', 'table', 'query', 'data', 'excessive', 'data table', 'large', 'data accessesqueries', 'accessesqueries', 'connection', 'indexing', 'pooling', 'client', 'number', 'degradation', 'large data', 'data queries', 'connection pooling']
18	1331, 1303, 1189	7	['isolation', 'rom', 'rom code', 'shared', 'shared resources', 'cpu instructions', 'cpu', 'instructions', 'hardware structures', 'securitycritical cpu', 'untrusted agents', 'circuitry sensors', 'circuitry', 'soc', 'hardware', 'access', 'clearing', 'execution', 'sensors', 'memory circuits']
19	1059, 1053	7	['documentation', 'product', 'proper documentation', 'platforms', 'components', 'runtime support', 'support component', 'component', 'support', 'thirdparty components', 'thirdparty', 'products', 'selfmodifying', 'selfmodifying code', 'specific runtime', 'runtime', 'functionality', 'documentation practices', 'understanding', 'automaticallygenerated']
20	908, 826, 459	7	['resource', 'resources', 'released', 'resource management', 'uninitialized', 'supporting', 'supporting resources', 'functionsmethods', 'temporary supporting', 'released resource', 'lowlevel functionsmethods', 'management', 'network', 'memory network', 'application server', 'connections', 'network connections', 'lowlevel', 'allocation', 'temporary']

21	278, 276	7	['permissions', 'files', 'file', 'objects', 'installation', 'installation process', 'object', 'process', 'installed', 'access', 'copying', 'files directories', 'directories', 'file permissions', 'unauthorized', 'temporary', 'copied', 'insecure permissions', 'installed files', 'permissions installation']
22	1241, 338	6	['numbers', 'random', 'generated', 'random number', 'prng', 'algorithm', 'number', 'pseudorandom', 'number generator', 'generator', 'pseudorandom numbers', 'random numbers', 'randomness', 'predictable', 'prng algorithm', 'generated random', 'generated numbers', 'entropy', 'cryptographic', 'generation']
23	1070, 1066, 375	6	['data', 'nonces', 'mutable data', 'mutable', 'serialization', 'serialized', 'function', 'serializable', 'member elements', 'elements', 'member', 'noncloned', 'serialized data', 'noncloned mutable', 'nonserializable', 'nonserializable member', 'manager component', 'designated', 'serializable storable', 'designated data']
24	1209, 1221, 1224	5	['hardware', 'hardware design', 'register', 'hardware module', 'module', 'control register', 'design control', 'design', 'language code', 'description language', 'hardware description', 'access hardware', 'bits', 'description', 'language', 'securitysensitive hardware', 'semiconductor defects', 'semiconductor', 'defects', 'register defaults']
25	1204, 1240	5	['cryptographic', 'algorithm', 'encrypted', 'nonstandard', 'ivs', 'cryptographic primitive', 'cryptographic implementation', 'inputs', 'primitive', 'cryptographic operations', 'hash', 'obfuscated encrypted', 'obfuscated', 'weaknesses', 'implementation', 'data', 'encrypted inputs', 'using nonstandard', 'case', 'using']
26	1087, 1045, 1079	5	['class', 'destructor', 'virtual', 'child', 'parent', 'leaks', 'parent class', 'child class', 'virtual destructor', 'memory leaks', 'classes', 'instance', 'deleted', 'memory', 'class instance', 'derived', 'objects', 'undefined behavior', 'undefined', 'leaks undefined']

27	353, 319	5	['transmission', 'transmitted', 'communication', 'message', 'channel', 'communication channel', 'data transmitted', 'data', 'securitycritical data', 'sensitive securitycritical', 'decoding', 'data transmission', 'network', 'decoding mechanism', 'securitycritical', 'unauthorized', 'case', 'integrity data', 'transmitted data', 'sniffing']
28	1291, 1243, 1295	5	['debug', 'debugging', 'debugging messages', 'production code', 'production', 'debug components', 'products debug', 'messages', 'fuses', 'securitysensitive information', 'debug functionality', 'information', 'debug mode', 'mode', 'stored fuses', 'components', 'access', 'securitysensitive', 'configuration protection', 'information stored']
29	1298, 1231, 1246	5	['hardware', 'wear', 'registers', 'wear leveling', 'memories', 'leveling', 'register', 'nonvolatile', 'nonvolatile memories', 'memorymapped io', 'memorymapped', 'leveling operations', 'io registers', 'writeonce register', 'software component', 'writeonce', 'io', 'limitedwrite nonvolatile', 'limitedwrite', 'hardware configuration']
30	439, 420, 454	5	['alternate', 'alternate channel', 'channel', 'network', 'initialization', 'data stores', 'variables data', 'changes', 'critical internal', 'internal variables', 'behavior functionality', 'stores', 'new', 'data', 'transmission', 'initialization critical', 'changes behavior', 'integrity', 'variables', 'predictable']
31	1050, 835, 1095	5	['loop', 'infinite', 'nonreentrant function', 'nonreentrant', 'loops', 'condition', 'infinite loop', 'loop conditions', 'concurrent', 'platform resources', 'excessive looping', 'looping', 'platform', 'function', 'resource exhaustion', 'exhaustion', 'loop condition', 'conditions', 'contention', 'resource']
32	798, 1392, 1273	5	['credentials', 'authentication', 'default credentials', 'default', 'hardcoded credentials', 'authentication algorithm', 'unauthorized', 'access', 'parties', 'unauthorized access', 'hardcoded', 'attackers', 'algorithm', 'sensitive', 'sensitive information', 'product', 'code configuration', 'threat', 'information', 'implementation']

33	1188, 708, 1051	4	['hardcoded values', 'network', 'owner', 'resources', 'values', 'hardcoded', 'default value', 'resource', 'default', 'network resources', 'insecure default', 'assignment', 'ownership', 'affected resources', 'unauthorized owner', 'control', 'insecure', 'access', 'unauthorized', 'value']
34	1114, 1115, 1113	4	['comments', 'code', 'source code', 'source', 'standards', 'inaccurate', 'inconsistent', 'inaccurate comments', 'inconsistent nonstandard', 'whitespace', 'code files', 'nonstandard', 'coding standards', 'comment styles', 'comment', 'misleading comments', 'headers', 'prologues headers', 'prologues', 'styles']
35	1328, 1283, 1326	4	['boot', 'firmware', 'process', 'boot process', 'secureboot', 'volatile memory', 'volatile', 'secureboot process', 'boot flow', 'secure boot', 'flow verification', 'securityversion', 'register contents', 'register', 'versions', 'hardware', 'securityversion number', 'flow', 'contents', 'verification process']
36	1341, 772, 771	4	['resource', 'resources', 'resource management', 'management', 'release', 'proper resource', 'handling resources', 'resources handles', 'handles', 'unique identifiers', 'needed', 'longer needed', 'released', 'allocated', 'improper handling', 'identifiers', 'leads', 'longer', 'unique', 'handling']
37	444, 941, 421	4	['channel', 'communication', 'http', 'request', 'alternate', 'requests', 'alternate channel', 'communication channel', 'malformed', 'outgoing', 'intermediary http', 'incoming', 'destination', 'intermediary', 'http requests', 'http agent', 'requests responses', 'client server', 'unauthorized', 'responses']
38	1304, 1281, 1272	4	['power', 'processor', 'reset', 'state', 'debug state', 'power debug', 'power saverestore', 'saverestore', 'processor instructions', 'securitycritical logic', 'instructions', 'configuration settings', 'configuration', 'settings', 'undesirable behavior', 'undesirable', 'logic', 'debug', 'securitycritical', 'state transitions']
39	832, 414, 412	4	['lock', 'resource', 'locked', 'locking', 'locked resource', 'lock checking', 'unlocking', 'proper lock', 'resource locked', 'excessive locking', 'checking', 'sensitive operations', 'unlock', 'mechanisms', 'proper', 'products', 'resources', 'lock performing', 'locking behavior', 'unlocking resource']

40	366, 821, 820	4	['synchronization', 'shared resource', 'shared', 'threads', 'deadlock', 'resource', 'multiple threads', 'multiple', 'threads processes', 'processes access', 'lack synchronization', 'proper synchronization', 'segments', 'concurrent', 'access shared', 'coordination', 'threat', 'synchronization mechanisms', 'execution', 'resource simultaneously']
41	1311, 1315, 1317	3	['fabric', 'transactions', 'ip blocks', 'translation', 'blocks', 'ip', 'data transferred', 'control', 'transferred', 'fabric transactions', 'fabric endpoint', 'conversion process', 'transactions fabric', 'devices', 'endpoint', 'incorrect translation', 'responder devices', 'control fabric', 'responder', 'transactions ip']
42	428, 427, 426	3	['path', 'search', 'search path', 'resources', 'controlled search', 'controlled', 'critical resources', 'files', 'resources accessed', 'critical', 'manipulating search', 'accessed', 'compromise', 'locations', 'malicious', 'attackers', 'configuration', 'configuration files', 'include', 'malicious code']
43	1108, 1063, 1126	3	['variables', 'variable', 'global', 'global variables', 'static', 'static code', 'code', 'scope', 'block', 'code block', 'object', 'creation', 'instance class', 'use global', 'declaration', 'code blocks', 'coding', 'larger', 'creating', 'instance']
44	1102, 1025, 1024	3	['comparison', 'data representation', 'representation', 'incorrect comparison', 'entities', 'incorrect', 'compared', 'entities compared', 'lowlevel data', 'comparisons', 'lowlevel', 'comparison entities', 'incompatible types', 'data', 'incompatible', 'manipulate entities', 'inconsistencies errors', 'characteristics', 'code', 'identifying instances']
45	1335, 681, 1339	3	['real', 'calculations', 'real numbers', 'accuracy', 'accuracy precision', 'precision', 'fractional', 'shifting', 'integer', 'conversion', 'numbers', 'values', 'data conversions', 'real number', 'conversions', 'software', 'unexpected', 'integer values', 'systems calculations', 'accurate calculations']
46	639, 1230, 921	3	['metadata', 'file device', 'access', 'access control', 'information', 'file', 'sensitive information', 'device', 'records', 'sensitive', 'authorization', 'control', 'key', 'unauthorized', 'key value', 'data', 'control mechanisms', 'value', 'mechanisms', 'individuals']

47	547, 1106, 1107	3	['constants', 'symbolic', 'hardcoded constants', 'definitions', 'constant definitions', 'constant', 'source code', 'code', 'hardcoded', 'source', 'literal constants', 'literal', 'symbolic constants', 'scattered', 'instead', 'constants instead', 'scattered constant', 'constants source', 'instead symbolic', 'centralized']
48	463, 477, 464	3	['sentinel', 'deprecated', 'functions', 'accidental', 'deprecated functions', 'problems', 'datastructure', 'logic problems', 'sentinel value', 'datastructure sentinel', 'deletion datastructure', 'accidental deletion', 'logic', 'software', 'obsolete functions', 'deprecated obsolete', 'data structure', 'codebase', 'deletion', 'programming']
49	419, 915, 914	3	['variables', 'attributes', 'dynamicallyidentified', 'dynamicallyidentified variables', 'restricted functionality', 'administration', 'attributes modified', 'administration restricted', 'product', 'functionality', 'restricted', 'variables involved', 'modified', 'primary channel', 'channel', 'modification', 'upstream', 'upstream component', 'unauthorized', 'primary']
50	59, 112	3	['xml', 'malicious xml', 'product', 'xml product', 'filename', 'case', 'malicious', 'untrusted source', 'file', 'implementing', 'filebased operations', 'link shortcut', 'schema', 'unintended resource', 'appropriate schema', 'filebased', 'processing malicious', 'virtual resource', 'xml case', 'shortcut']
51	1037, 1301	2	['removal', 'securitycritical protection', 'data removal', 'removal process', 'hardware components', 'securitycritical', 'protection', 'optimization', 'potentially sensitive', 'protection mechanism', 'data', 'components', 'software', 'information', 'processors optimization', 'optimization program', 'removal modification', 'removed modified', 'sensitive information', 'process']
52	1222, 1232	2	['addresses', 'protected', 'lock', 'configuration', 'protected registers', 'registers lock', 'lock bits', 'power state', 'state transitions', 'transitions', 'protected addresses', 'conflict', 'bit', 'bits', 'registers', 'configuration lock', 'conflict functional', 'programmability', 'power', 'configuration changes']

53	475, 648	2	['function', 'api requirements', 'requirements', 'parameter', 'api', 'undefined behavior', 'control parameter', 'undefined', 'application', 'privileges', 'behavior', 'behavior function', 'vulnerable function', 'utilizing vulnerable', 'potentially', 'potentially confidentiality', 'elevated privileges', 'application utilizing', 'monitoring abnormal', 'perform unauthorized']
54	1312, 1260	2	['regions', 'memory', 'mmio regions', 'mmio', 'mirrored memory', 'mirrored', 'memory areas', 'protection', 'overlapping', 'memory regions', 'memory protection', 'memory mmio', 'protected memory', 'firewall', 'areas', 'unprotected', 'protected', 'address regions', 'address', 'protection mechanisms']
55	348, 501	2	['trusted', 'data', 'untrusted data', 'data sources', 'sources', 'trusted untrusted', 'untrusted', 'mixed', 'structured', 'trusted data', 'trust', 'structured message', 'structure structured', 'verification trust', 'mixed data', 'validation segregation', 'verification', 'segregation', 'data structure', 'structure']
56	1351, 1314	2	['sensor', 'sensor data', 'device', 'data values', 'parametric', 'readings', 'parametric data', 'untrusted software', 'temperatures', 'threat', 'sensors', 'standard', 'operational failure', 'writeprotection', 'standard operating', 'values', 'hardware', 'untrusted', 'operational', 'accurate']

TABLE C.1: A table indicating the ID, names... etc of the initial 57 clusters from the topic modelling stage, for the selected model configuration (ID 4).

ID	Rep. CWEs	Parent rep CWEs	Threat description
-1	1298, 641, 843		{outliers}
0	125, 140, 76	119, 75, 138, 20, 707 (pillar)	Missing or improper neutralisation
1	790, 78, 74	77, 707 (pillar), 20	Unfiltered special elements
2	1287, 1173, 625	20, 185	Improper input validation
3	1125, 203, 1269	1120, 200, 693	Insecure product configuration
4	824, 466, 788	119	Insecure memory buffer (i.e. buffer overflow)
5	488, 613, 425	668, 672, 862, 424	Improper session management (i.e. web authorisation)
6	549, 309, 916	1391, 1390	Weak implementation or configuration of password-based authentication
7	521, 640, 620	1391, 1390	Weak implementation or configuration of password-based authentication
8	394, 440, 393	684, 754	Improper adherence to coding standards (i.e. exception handling)
9	270, 267, 653	269, 693, 657	Insufficient privilege management or compartmentalisation
10	1300, 1192, 1319	203, 657, 693	Side channel and SoC threats
11	804, 1220	863, 1390, 284	Weak or misconfigured authentication and authorisation
12	544, 396, 397	755, 705, 221, 703	Missing or poor error handling
13	1122, 1124, 1121	1120	Excessive code complexity
14	322, 325	306, 573	Missing authentication
15	356, 448, 449	355	Error or feature omission in UI
16	1118, 1110, 1111	1059	Insufficient technical documentation
17	1072, 1049, 1089	1006	Bad database coding practices
18	1331, 1303, 1189	668, 653, 203, 1189	Improper sharing of architectural hardware resources (NoC, SoC etc)
19	1059, 1053	710, 1059	Insufficient technical documentation
20	908, 826, 459	665, 666, 404, 665 (pillar)	Improper control of resource through its lifetime
21	278, 276	732	Incorrect permission assignment for critical resource

22	1241, 338	330	Use of insufficiently random values
23	1070, 1066, 375	710, 668	Bad software coding practices
24	1209, 1221, 1224	710, 665, 284	Bad hardware coding practices
25	1204, 1240	330, 327, 693	Improperly configured or weak encryption algorithm
26	1087, 1045, 1079	1076	Improper adherence to coding standards (virtual methods)
27	353, 319	345, 311	Missing or insufficient encryption (incl. integrity checks)
28	1291, 1243, 1295	1207	Exposure of sensitive debug messages
29	1298, 1231, 1246	1199, 1202	Circuit, logic and memory weaknesses
30	439, 420, 454	363, 923, 665	Hardware logic weaknesses (incl. side channel)
31	1050, 835, 1095	405, 1006, 834, 438, 1120, 1226	Race condition and resource consumption
32	798, 1392, 1273	1391, 200	Improper or weak credential management
33	1188, 708, 1051	665, 282	Insecure default or hardcoded configuration
34	1114, 1115, 1113	1078	Inappropriate source code style or formatting
35	1328, 1283, 1326	285, 284, 693 (pillar)	Mutable hardware/firmware registers by unauthorised users
36	1341, 772, 771	675, 404, 400, 664 (pillar)	Improper control of resource through its lifetime
37	444, 941, 421	436, 923, 362	Communication or configuration conflict between endpoints
38	1304, 1281, 1272	284, 691, 226	Leakage during hardware state transition
39	832, 414, 412	667	Improper locking
40	366, 821, 820	362, 662	Improper syncronisation using shared resources
41	1311, 1315, 1317	284	Improper access control in fabric bridge (between hardware end-points)
42	428, 427, 426	668, 673, 642	Violation of a trust boundary
43	1108, 1063, 1126	1076, 1176, 710 (pillar)	Bad software coding practices
44	1102, 1025, 1024	758, 697 (pillar)	Improper comparisons of data (representation, types, methods)
45	1335, 681, 1339	682 (pillar), 704	Incorrect or improper calculation

46	639, 1230, 921	863, 285, 922	Insufficient protection of sensitive information
47	547, 1106, 1107	1078	Inappropriate source code style or formatting
48	463, 477, 464	707 (pillar), 710 (pillar), 138	Improper adherence to coding standards (accidents and maintenance)
49	419, 915, 914	923, 913,	Improperly controlled dymanimcally-managed resources
50	59, 112	706, 1286	Improper input validation (web)
51	1037, 1301	1038	Improper adherence to coding standards (optimisation)
52	1222, 1232	1220, 667	Improper locking (access control granularity)
53	475, 648	573, 269	Improper adherence to coding standards
54	1312, 1260	284	Improper access control (hardware)
55	348, 501	345, 664	Violation of a trust boundary
56	1351, 1314	1384, 862	Circuit, logic and memory weaknesses

TABLE C.2: A table describing some of the topic properties used during the cluster merging process, including the topic ID, the representative CWEs, the parent CWEs of those representative CWEs and the threat descriptions given to each topic, for the selected model configuration (ID 4). In addition to these, the hierarchical clustering diagram and the similarity matrix were used, alongside other qualitative considerations.

ID	Cluster merged into	Merged threat description	Merged count	Original count
-1		{outliers}		136
0		Missing or improper neutralisation	54	35
1	0			19
2		Improper input validation	18	15
3		Insecure product configuration	18	14
4		Insecure memory buffer	14	14
5		Improper session management	14	10
6		Weak implementation or use of password-based authentication	20	10
7	6			10
8		Improper adherence to coding best practice	69	9
9		Insufficient privilege management or compartmentalisation	14	9
10		Side channel and SoC threats	14	9
11		Insufficient or misconfigured access control	12	9
12	8			8
13	8			8
14		Missing authentication	8	8
15		Error or feature omission in UI	7	7
16		Insufficient technical documentation	14	7
17		Improper adherence to database best practice	7	7
18		Deficient hardware architecture/security features	32	7
19	16			7
20		Improper control of software resource through its lifetime	24	7
21		Incorrect permission assignment for critical resource	7	7

22		Improperly configured or weak encryption	16	6
23	8			6
24	18			5
25	22			5
26	8			5
27	22			5
28	8			5
29	18			5
30	10			5
31	8			5
32		Improper or weak credential management	5	5
33	3			4
34	8			4
35	18			4
36	20			4
37	5			4
38	18			4
39	20			4
40	20			4
41	18			3
42	9			3
43	8			3
44	8			3
45	8			3

46	11			3
47	8			3
48	8			3
49	20			3
50	2			3
51	8			2
52	20			2
53	8			2
54	18			2
55	9			2
56	18			2

TABLE C.3: A table indicating the results and decision-making behind the merging process, including the topic ID, the ID of the parent topic to which child clusters were merged into, the merged topic description, the merged cluster count and the original cluster count, for the selected model configuration (ID 4).

References

Abdulmajeed Alahmari and Bob Duncan. Cybersecurity Risk Management in Small and Medium-Sized Enterprises: A Systematic Review of Recent Evidence. In *2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, Cyber SA 2020*. Institute of Electrical and Electronics Engineers Inc., 6 2020. ISBN 9781728166902. .

Dimo Angelov. Top2Vec: Distributed Representations of Topics. 8 2020. URL https://arxiv.org/abs/2008.09470v1.

Elchin Asgarli and Eric Burger. Semantic ontologies for cyber threat sharing standards. In *2016 IEEE Symposium on Technologies for Homeland Security, HST 2016*. Institute of Electrical and Electronics Engineers Inc., 9 2016. ISBN 9781509007707. .

M. Atighetchi, Borislava I. Simidchieva, F. Yaman, T. Eskridge, M. Carvalho, and Nicholas Paltzer. Using Ontologies to Quantify Attack Surfaces. *STIDS*, 2016.

Adiel Aviad and Krzysztof Węcel. Cyber Treat Intelligence Modeling. In *Lecture Notes in Business Information Processing*, volume 353, pages 361–370. Springer Verlag, 6 2019. ISBN 9783030204846. . URL https://doi.org/10.1007/978-3-030-20485-3_28.

B. Schneier. Attack Trees - Schneier on Security, 1999. URL https://www.schneier.com/academic/archives/1999/12/attack_trees.html.

Maria Bada, Angela M. Sasse, and Jason R. C. Nurse. Cyber Security Awareness Campaigns: Why do they fail to change behaviour? 1 2019. URL http://arxiv.org/abs/1901.02672.

Ghattas Badih, Michel Pierre, and Boyer Laurent. Assessing variable importance in clustering: a new method based on unsupervised binary decision trees. *Computational Statistics*, 34(1):301–321, 3 2019. ISSN 16139658. . URL https://link.springer.com/article/10.1007/s00180-018-0857-0.

Guilherme Baesso Moreira, Vanusa Menditi Calegario, Julio Cesar Duarte, and Anderson F. Pereira Dos Santos. Extending the VERIS Framework to an Incident Handling

Ontology. In *Proceedings - 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018*, pages 440–445. Institute of Electrical and Electronics Engineers Inc., 1 2019. ISBN 9781538673256. .

Yves Barlette, Katherine Gundolf, and Annabelle Jaouen. CEOs' information security behavior in SMEs: Does ownership matter? *Post-Print*, 2017. URL https://ideas.repec.org/p/hal/journl/hal-02000435.html.

Sean Barnum. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression. 2014. URL https://stixproject.github.io/getting-started/whitepaper/.

V. Basili, G. Caldiera, and H. D. Rombach. The Goal Question Metric Approach. *Encyclopedia of software engineering*, 1994.

Anass Bayaga, Stephen Flowerday, and Liezel Cilliers. IT Risk and Chaos Theory: Effect on the performance of South African SMEs. In *WMSCI*, Florida, USA, 2017. URL https://www.researchgate.net/publication/322701209_IT_Risk_and_Chaos_Theory_Effect_on_the_performance_of_South_African_SMEs.

Sussy Bayona-Oré, Jose A. Calvo-Manzano, Gonzalo Cuevas, and Tomas San-Feliu. Critical success factors taxonomy for software process deployment. *Software Quality Journal*, 22(1):21–48, 3 2014. ISSN 09639314. . URL https://link.springer.com/article/10.1007/s11219-012-9190-y.

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 5 2001. ISSN 00368733. . URL https://www.scientificamerican.com/article/the-semantic-web/.

Carlos Blanco, Joaquín Lasheras, Rafael Valencia-García, Eduardo Fernández-Medina, Ambrosio Toval, and Mario Piattini. A systematic review and comparison of security ontologies. In *ARES 2008 - 3rd International Conference on Availability, Security, and Reliability, Proceedings*, pages 813–820, 2008. ISBN 0769531024. .

Carlos Blanco, Joaquín Lasheras, Eduardo Fernández-Medina, Rafael Valencia-García, and Ambrosio Toval. Basis for an integrated security ontology according to a systematic review of existing proposals. *Computer Standards & Interfaces*, 33(4):372–388, 6 2011. ISSN 0920-5489. .

David M Blei, Andrew Y Ng, and Jordan@cs Berkeley Edu. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 3 2003. . URL https://dl.acm.org/doi/10.5555/944919.944937.

G. Bouma. Normalized (pointwise) mutual information in collocation extraction. 2009.

Virginia Braun and Victoria Clarke. Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006. ISSN 14780887. .

Maricela Bravo, Luis Fernando Hoyos Reyes, and José A. Reyes Ortiz. Methodology for ontology design and construction. *Contaduria y Administracion*, 64(4):1–24, 2019. ISSN 24488410. .

Siri Bromander, A. Jøsang, and Martin Eian. Semantic Cyberthreat Modelling. *STIDS*, 2016.

Siri Bromander, Lilly Pijnenburg Muller, Martin Eian, and Audun Jøsang. Examining the "Known Truths" in Cyber Threat Intelligence-The Case of STIX. 2020. .

Luben M.C. Cabezas, Rafael Izbicki, and Rafael B. Stern. Hierarchical clustering: visualization, feature importance and model selection. *Applied Soft Computing*, 141, 11 2021. ISSN 15684946. . URL https://arxiv.org/abs/2112.01372v2.

T. Caliński and J. Harabasz. A Dendrite Method Foe Cluster Analysis. *Communications in Statistics*, 3(1):1–27, 1974. ISSN 00903272. .

S. Caltagirone, Andy Pendergast, and Chris Betz. The Diamond Model of Intrusion Analysis. Technical report, Centre for Cyber Intelligence Analyis and Threat Research, Hannover MD, 2013.

Richard A Caralli, James F Stevens, Lisa R Young, and William R Wilson. Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process. Technical report, 2007. URL http://www.sei.cmu.edu/publications/pubweb.html.

Chris Ensor. A brief history of Cyber Essentials, 2017. URL https://webarchive.nationalarchives.gov.uk/20190902112835/https://www.cyberessentials.ncsc.gov.uk/2017/11/27/a-brief-history-of-cyber-essentials.

Chun Jen Chung, Pankaj Khatkar, Tianyi Xing, Jeongkeun Lee, and Dijiang Huang. NICE: Network intrusion detection and countermeasure selection in virtual network systems. *IEEE Transactions on Dependable and Secure Computing*, 10(4):198–211, 2013. ISSN 15455971. .

Robert Colomb. Quality of ontologies in interoperating information systems. 2002. URL http://www.opengrey.eu/item/display/10068/305439.

Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. Where is their meeting point?, 2003. ISSN 0169023X.

J. W Creswell and Creswell J. D. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage Publishing, 5th edition edition, 2017. URL https://us.sagepub.com/en-us/nam/research-design/book255675.

DCMS. Cyber Security Breaches Survey 2023. Technical report, 2023. URL https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2023/cyber-security-breaches-survey-2023.

Department for Business Energy and Industrial Strategy. Business population estimates for the UK and regions 2024. URL https://www.gov.uk/government/statistics/business-population-estimates-2024/business-population-estimates-for-the-uk-and-regions-2024-statistical-release.

Department for Business Innovation & Skills. Cyber security boost for UK firms , 1 2015. URL https://www.gov.uk/government/news/cyber-security-boost-for-uk-firms.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1: 4171–4186, 10 2018. URL https://arxiv.org/abs/1810.04805v2.

Adji B. Dieng, Francisco J.R. Ruiz, and David M. Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 1 2020. ISSN 2307387X. . URL https://dx.doi.org/10.1162/tacl_a_00325.

Tharam S. Dillon and Poh L. Tan. *Object-Oriented Conceptual Modeling*. 1993. URL https://dl.acm.org/doi/10.5555/562767.

F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in Description Logics. *Principles of knowledge representation*, pages 191–236, 1997. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.4180.

Roman Egger and Joanne Yu. A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts. *Frontiers in Sociology*, 7:886498, 5 2022. ISSN 22977775. . URL /pmc/articles/PMC9120935//pmc/articles/PMC9120935/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC9120935/.

N S Egoshin, A A Konev, and A A Shelupanov. A model of threats to the confidentiality of information processed in cyberspace based on the information flows model. *Symmetry*, 12(11):1–18, 2020. ISSN 20738994. .

Charles A. Ellis, Mohammad S. E. Sendi, Eloy P. T. Geenjaar, Sergey M. Plis, Robyn L. Miller, and Vince D. Calhoun. Algorithm-Agnostic Explainability for Unsupervised Clustering. 5 2021. URL https://arxiv.org/abs/2105.08053v2.

Isra Elsharef, Zhen Zeng, and Zhongshu Gu. Facilitating Threat Modeling by Leveraging Large Language Models - NDSS Symposium. In *NDSS*, 2024. URL https://www.ndss-symposium.org/ndss-paper/auto-draft-539/.

European Commission. ANNUAL REPORT ON EUROPEAN SMEs 2018/2019, 2019. URL https://ec.europa.eu/growth/smes/business-friendly-environment/performance-review_en.

Nawfal Fadhel, Federico Lombardi, Leonardo Aniello, Andrea Margheri, and Vladimiro Sassone. Towards a semantic modelling for threat analysis of IoT applications: A case study on transactive energy. In *IET Conference Publications*, volume 2019, pages 22 (6 pp.)–22 (6 pp.). Institution of Engineering and Technology, 2019. ISBN 9781839530074. . URL https://digital-library.theiet.org/content/conferences/10.1049/cp.2019.0147.

Eva Maria Falkner and Martin R.W. Hiebl. Risk management in SMEs: a systematic review of available evidence, 3 2015. ISSN 09657967.

Christina Feilmayr and Wolfram Wöß. An analysis of ontologies and their success factors for application to business. *Data and Knowledge Engineering*, 101:1–23, 1 2016. ISSN 0169023X. .

Stefan Fenz. Ontology-based generation of IT-security metrics. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1833–1839, 2010. ISBN 9781605586380. .

Stefan Fenz and Andreas Ekelhart. Formalizing information security knowledge. In *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security, ASIACCS'09*, pages 183–194, New York, New York, USA, 2009. ACM Press. ISBN 9781605583945. . URL http://portal.acm.org/citation.cfm?doid=1533057.1533084.

Eduardo B. Fernandez and Raul Monge. A security reference architecture for cloud systems. In *Proceedings of the First International Conference on Dependable and Secure Cloud Computing Architecture - DASCCA '14*, New York, New York, USA, 2014. ACM Press. ISBN 9781450325233. .

Eduardo B. Fernandez, Raul Monge, and Keiko Hashizume. Building a security reference architecture for cloud systems. *Requirements Engineering*, 21(2):225–249, 6 2016. ISSN 1432010X. .

Jesualdo Tomás Fernández-Breis and Rodrigo Martínez-Béjar. A cooperative framework for integrating ontologies. *International Journal of Human Computer Studies*, 56 (6):665–720, 2002. ISSN 10715819. .

Eduardo Fernandez-Buglioni. *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns [Book]*. Wiley, 2013. URL https://www.oreilly.com/library/view/security-patterns-in/9781119970484/.

M. Fernández-López, Asunción Gómez-Pérez, and N. Juristo. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. *AAAI*, 1997.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20, 1 2018. ISSN 15337928. URL https://arxiv.org/abs/1801.01489v5.

Franz Baader, Diego Calvanese, Deborah L Mcguinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. 2 2007. URL https://www.researchgate.net/publication/230745455_The_Description_Logic_Handbook_Theory_Implementation_and_Applications.

Aldo Gangemi and Valentina Presutti. Ontology Design Patterns. In *Handbook on Ontologies*, pages 221–243. Springer Berlin Heidelberg, 2009. .

Jian Bo Gao, Bao Wen Zhang, Xiao Hua Chen, and Zheng Luo. Ontology-based model of network and computer attacks for security assessment. *Journal of Shanghai Jiaotong University (Science)*, 18(5):554–562, 10 2013. ISSN 10071172. . URL https://www.researchgate.net/publication/269381790_Ontology-based_model_of_network_and_computer_attacks_for_security_assessment.

Dragan Gaševic, Dragan Djuric, and Vladan Devedžic. *Model Driven Engineering and Ontology Development*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. . URL http://link.springer.com/10.1007/978-3-642-00282-3_1.

L. R. Gay. *Educational research: competencies for analysis and application*. Prentice-Hall, 10th edition edition, 1996.

Michael Genesereth and Nils Nilsson. *Logical Foundations of Artificial Intelligence - 1st Edition*. 1987. URL https://www.elsevier.com/books/logical-foundations-of-artificial-intelligence/genesereth/978-0-934613-31-6.

Nazila Gol Mohammadi, Torsten Bandyszak, Micha Moffie, Xiaoyu Chen, Thorsten Weyer, Costas Kalogiros, Bassem Nasser, and Mike Surridge. Maintaining trustworthiness of socio-technical systems at run-time. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8647 LNCS:1–12, 2014. .

Saptarsi Goswami, Amlan Chakrabarti, and Basabi Chakraborty. An efficient feature selection technique for clustering based on a new measure of feature importance. *Journal of Intelligent & Fuzzy Systems*, 32(6):3847–3858, 1 2017. ISSN 1064-1246. .

Thomas R Gruber. A Translation Approach to Portable Ontology Specifications A Translation Approach to Portable Ontology Specifications. Technical report, 1993.

Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human - Computer Studies*, 43(5-6):907–928, 1995. ISSN 10959300. .

M. Gruninger. Methodology for the Design and Evaluation of Ontologies. *IJCAI*, 1995.

Michael Gruninger and Jintae Lee. Ontology Applications and Design. *Communications of the ACM*, 45(2):39–41, 2 2002. ISSN 0001-0782. .

Nicola Guarino. Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1299, pages 139–170. Springer Verlag, 1997. ISBN 354063438X. .

Nicola Guarino, Daniel Oberle, and Steffen Staab. What Is an Ontology? In *Handbook on Ontologies*, pages 1–17. Springer Berlin Heidelberg, 2009. .

Tapiwa Gundu. Acknowledging and Reducing the Knowing and Doing Gap in Employee Cybersecurity Compliance. In *Conference: International Conference on Cyber Warfare and Security*, Stellenbosch, 2019. URL https://www.researchgate.net/publication/333044935_Acknowledging_and_Reducing_the_Knowing_and_Doing_Gap_in_Employee_Cybersecurity_Compliance.

Amelie Gyrard, Christian Bonnet, and Karima Boudaoud. The STAC (security toolbox: Attacks & countermeasures) ontology. In *22nd International Conference on World Wide Web*, pages 165–166. Association for Computing Machinery, 2013. ISBN 9781450320382. . URL https://www.researchgate.net/publication/262177689_The_STAC_Security_Toolbox_Attacks_Countermeasures_ontology.

Maja Hadzic, Pornpit Wongthongtham, Tharam Dillon, and Elizabeth Chang. Ontology design approaches. *Studies in Computational Intelligence*, 219:75–91, 2009. ISSN 1860949X. . URL https://link.springer.com/chapter/10.1007/978-3-642-01904-3_5.

M Hamad and V Prevelakis. SAVTA: A hybrid vehicular threat model: Overview and case study. *Information (Switzerland)*, 11(5), 2020. ISSN 20782489. .

Stephen Hart, Andrea Margheri, Federica Paci, and Vladimiro Sassone. Riskio: A Serious Game for Cyber Security Awareness and Education. *Computers and Security*, 95: 101827, 8 2020. ISSN 01674048. .

Chad D. Heitzenrater and Andrew C. Simpson. Policy, statistics and questions: Reflections on UK cyber security disclosures. *Journal of Cybersecurity*, 2(1):43–56, 12 2016. ISSN 20572093. . URL http://cyberessentials.org/background/.

Amit Hendre and Karuna Pande Joshi. A Semantic Approach to Cloud Security and Compliance. In *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015*, pages 1081–1084. Institute of Electrical and Electronics Engineers Inc., 8 2015. ISBN 9781467372879. .

Almut Herzog, Nahid Shahmehri, and Claudiu Duma. An Ontology of Information Security. *International Journal of Information Security and Privacy (IJISP)*, 1(4):1–23, 2007. ISSN 19301669. .

Md Imran Hossen, Ashraful Islam, Farzana Anowar, Eshtiak Ahmed, Mohammad Masudur Rahman, Xiali, and Hei. Generating Cyber Threat Intelligence to Discover Potential Security Threats Using Classification and Topic Modeling. 8 2021. URL https://arxiv.org/abs/2108.06862v3.

Eric M Hutchins, Michael J Cloppert, and Rohan M Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. 2 2011. URL https://www.researchgate.net/publication/266038451_Intelligence-Driven_Computer_Network_Defense_Informed_by_Analysis_of_Adversary_Campaigns_and_Intrusion_Kill_Chains.

Terrance R Ingoldsby. Attack Tree-based Threat Risk Analysis. Technical report, 2009. URL www.amenaza.com.

Oumaima Alaoui Ismaili, Vincent Lemaire, and Antoine Cornuéjols. A supervised methodology to measure the variables contribution to a clustering. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8834:159–166, 2014. ISSN 16113349. . URL https://link.springer.com/chapter/10.1007/978-3-319-12637-1_20.

M Jbair, B Ahmad, C Maple, and R Harrison. Threat modelling for industrial cyber physical systems in the era of smart manufacturing. *Computers in Industry*, 137, 2022. ISSN 01663615. . URL https://www.sciencedirect.com/science/article/abs/pii/S0166361522000069.

I. T Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2002. ISBN 0-387-95442-2. . URL http://link.springer.com/10.1007/b98835.

M Joshi, B Mazumdar, and S Dey. A comprehensive security analysis of match-in-database fingerprint biometric system. *Pattern Recognition Letters*, 138:247–266, 2020. ISSN 01678655. .

Jan Jürjens. UMLsec: Extending UML for secure systems development. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2460 LNCS, pages 412–425. Springer Verlag, 2002. ISBN 3540442545. . URL https://link.springer.com/chapter/10.1007/3-540-45800-X_32.

Salah Kabanda, Maureen Tanner, and Cameron Kent. Exploring SME cybersecurity practices in developing countries. *Journal of Organizational Computing and Electronic Commerce*, 28(3):269–282, 7 2018. ISSN 10919392. .

Christos Kalloniatis, Haralambos Mouratidis, Manousakis Vassilis, Shareeful Islam, Stefanos Gritzalis, and Evangelia Kavakli. Towards the design of secure and privacy-oriented information systems in the cloud: Identifying the major concepts. *Computer Standards & Interfaces*, 36(4), 6 2014. ISSN 09205489. .

Jasber Kaur and Norliana Mustafa. Examining the effects of knowledge, attitude and behaviour on information security awareness: A case on SME. In *International Conference on Research and Innovation in Information Systems, ICRIIS*, pages 286–290, 2013. ISBN 9781479924875. .

Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. STRIDE-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe 2017 - Proceedings*, volume 2018-January, pages 1–6. Institute of Electrical and Electronics Engineers Inc., 7 2017. ISBN 9781538619537. .

Margot Kimura, Troy Robert DeVries, and Susanna P. Gordon. The Cyber Defense (CyDef) Model for Assessing Countermeasure Capabilities. Technical report, Sandia National Laboratories (SNL), Sandia National Laboratory, 6 2017. URL http://www.osti.gov/servlets/purl/1367477/.

Barbara Kitchenham and Stuart M. Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering — BibSonomy. Technical report, Keele University and Durham University Joint Report, 7 2007. URL https://www.bibsonomy.org/bibtex/aed0229656ada843d3e3f24e5e5c9eb9.

Farzan Kolini. Clustering and Topic Modelling : A New Approach for Analysis of National Cybersecurity Strategies Completed Research Paper. 2017.

Igor Kotenko, Olga Polubelova, Igor Saenko, and Elena Doynikova. The ontology of metrics for security evaluation and decision support in SIEM systems. In *Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013*, pages 638–645, 2013. ISBN 9780769550084. .

Barbara H. Kwasnik. The Role of Classification in Knowledge Representation and Discovery. *Library trends*, 48(1), 6 1999. URL https://www.researchgate.net/publication/32961811_The_Role_of_Classification_in_Knowledge_Representation_and_Discovery.

Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi. A taxonomy of computer program security flaws. *ACM Computing Surveys*, 26(3):211–254, 9 1994. ISSN 0360-0300. .

Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *31st International Conference on Machine Learning, ICML 2014*, 4:2931–2939, 5 2014. URL https://arxiv.org/abs/1405.4053v2.

X Li, K He, Z Feng, and G Xu. Unified threat model for analyzing and evaluating software threats. *Security and Communication Networks*, 7(10):1454–1466, 2014. ISSN 19390114. .

Raymond Lister. Mixed methods. *ACM SIGCSE Bulletin*, 37(4):18–19, 12 2005. ISSN 0097-8418. . URL https://dl.acm.org/doi/10.1145/1113847.1113857.

F. Lopez. Overview Of Methodologies For Building Ontologies. *IJCAI 1999*, 1999.

Mariana Fernández López, Asunción Gómez-Pérez, Juan Pazos Sierra, and Alejandro Pazos Sierra. Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems and Their Applications*, 14(1):37–46, 1999. ISSN 10947167. .

Adolfo Lozano-Tello and Asunción Gómez-Pérez. ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management*, 15(2):1–18, 2004. ISSN 10638016. .

Scott M. Lundberg and Su In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 2017-December:4766–4775, 5 2017. ISSN 10495258. URL https://arxiv.org/abs/1705.07874v2.

Alexander Maedche and Steffen Staab. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79, 2001. ISSN 15411672. .

Vasileios Mavroeidis and Siri Bromander. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In *Proceedings - 2017 European Intelligence and Security Informatics Conference, EISIC 2017*, volume 2017-January, pages 91–98. Institute of Electrical and Electronics Engineers Inc., 12 2017. ISBN 9781538623855. .

J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Readings in nonmonotonic reasoning*, pages 26–45, 10 1987. URL https://dl.acm.org/doi/10.5555/42641.42642.

Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205, 3 2017. ISSN 2475-9066. . URL https://joss.theoj.org/papers/10.21105/joss.00205.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2 2018. URL https://arxiv.org/abs/1802.03426v3.

Rob McMillan. Definition: Threat Intelligence, 5 2013. URL https://www.gartner.com/en/documents/2487216.

Nancy R. Mead. Security Quality Requirements Engineering (SQUARE). *Software Engineering Institute*, 2005. URL https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=484884.

Florian Menges and Günther Pernul. A comparative analysis of incident reporting formats. *Computers and Security*, 73:87–101, 3 2018. ISSN 01674048. .

Jan Meszaros and Alena Buchalcevova. Introducing OSSF: A framework for online service cybersecurity risk management. *COMPUTERS & SECURITY*, 65:300–313, 3 2017. ISSN 0167-4048. .

David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew Mccallum. Optimizing Semantic Coherence in Topic Models, 2011. URL https://aclanthology.org/D11-1024/.

MITRE. CWE - Common Weakness Enumeration, a. URL https://cwe.mitre.org/.

MITRE. Common Attack Pattern Enumeration and Classification (CAPEC), b. URL https://capec.mitre.org/.

MITRE. Common Platform Enumeration (CPE), c. URL https://cpe.mitre.org/.

MITRE. Common Vulnerabilities and Exposures (CVE), d. URL https://cve.mitre.org/.

MITRE. Common Weakness Risk Analysis Framework (CWRAF), e. URL https://cwe.mitre.org/cwraf/.

MITRE. Common Weakness Scoring System (CWSS), f. URL https://cwe.mitre.org/cwss/cwss_v1.0.1.html.

MITRE. MAEC Project Documentation, g. URL https://maecproject.github.io/about-maec/.

MITRE. MITRE ATT&CK®, h. URL https://attack.mitre.org/.

Andrea Montemaggio, Stefano Iannucci, Tanmay Bhowmik, and John Hamilton. Designing a Methodological Framework for the Empirical Evaluation of Self-Protecting Systems. In *Proceedings - 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion, ACSOS-C 2020*, pages 218–223. Institute of Electrical and Electronics Engineers Inc., 8 2020. ISBN 9781728184142. .

NCSC. Cyber security for your organisation starts here, 2020. URL https://www.ncsc.gov.uk/cyberessentials/advice.

Surya Nepal, Miranda Zhang, Rajiv Ranjan, Armin Haller, and Dimitrios Georgakopoulos. An Ontology-based System for Cloud Infrastructure Services' Discovery. In *Proceedings of the 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*. IEEE, 2012. ISBN 978-1-936968-36-7. .

NIST. Common Vulnerability Scoring System (CVSS), a. URL https://www.first.org/cvss/.

NIST. National Vulnerability Database (NVD), b. URL https://nvd.nist.gov/.

NIST. Security Content Automation Protocol, c. URL https://csrc.nist.gov/projects/security-content-automation-protocol.

Natalya F Noy and Deborah L Mcguinness. Ontology Development 101: A Guide to Creating Your First Ontology. 2 2001. URL https://www.researchgate.net/publication/243772462_Ontology_Development_101_A_Guide_to_Creating_Your_First_Ontology.

OASIS CTI. Introduction to STIX, a. URL https://oasis-open.github.io/cti-documentation/stix/intro.

OASIS CTI. Introduction to TAXII, b. URL https://oasis-open.github.io/cti-documentation/taxii/intro.

L. Obrst, P. Chase, and R. Markeloff. Developing an Ontology of the Cyber Security Domain. *STIDS*, 2012.

A. Oltramari, L. Cranor, Robert J. Walls, and P. McDaniel. Building an Ontology of Cyber Security. *STIDS*, 2014.

Emma Osborn. Business versus Technology: Sources of the Perceived Lack of Cyber Security in SMEs. Technical report, 2014.

Emma Osborn and Andrew Simpson. Risk and the Small-Scale Cyber Security Decision Making Dialogue - A UK Case Study. *Computer Journal*, 61(4):472–495, 4 2018. ISSN 14602067. .

Emmanouil Panaousis, Andrew Fielder, Pasquale Malacaria, Chris Hankin, and Fabrizio Smeraldi. Cybersecurity games and investments: A decision support approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8840:266–286, 2014. ISSN 16113349. .

Simon Parkin, Andrew Fielder, and Alex Ashby. Pragmatic security: Modelling it security management responsibilities for SME archetypes. In *MIST 2016 - Proceedings of the International Workshop on Managing Insider Security Threats, co-located with CCS 2016*, pages 69–80, New York, New York, USA, 10 2016. Association for Computing Machinery, Inc. ISBN 9781450345712. . URL http://dl.acm.org/citation.cfm?doid=2995959.2995967.

Paul Saitta, Brenda Larcom, and Michael Eddington. Trike v. 1 Methodology Document [Draft]. Technical report, 2005. URL https://www.researchgate.net/publication/229046782_Trike_v_1_Methodology_Document_Draft.

Paweł Pawliński, Przemysław Jaroszewski, Piotr Kijewski, Łukasz Siewierski, Paweł Jacewicz, Przemysław Zielony, and Radosław Żuber. Actionable information for security incident response — ENISA. Technical report, ENISA, 2015. URL https://www.enisa.europa.eu/publications/actionable-information-for-security.

Fabian Pedregosa, Vincent Michel, Olivier Grisel OLIVIERGRISEL, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Jake Vanderplas, David Cournapeau, Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Bertrand Thirion, Olivier Grisel, Vincent Dubourg, Alexandre Passos, Matthieu Brucher, Matthieu Perrot andÉdouardand, andÉdouard Duchesnay, and FRÉdouard Duchesnay EDOUARDDUCHESNAY. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Marcus Pendleton, Richard Garcia-Lebron, Jin Hee Cho, and Shouhuai Xu. A survey on systems security metrics. *ACM Computing Surveys*, 49(4), 12 2016. ISSN 15577341. . URL http://dx.doi.org/10.1145/3005714.

Xuena Peng and Hong Zhao. An "attacker centric" cyber attack behavior analysis technique. In *International Conference on Advanced Communication Technology, ICACT*, volume 3, pages 2113–2117, 2007. .

H. Pinto, Steffen Staab, and Christoph Tempich. DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies. *undefined*, 2004.

D. M. Pisanelli, A. Gangemi, and G. Steve. Ontologies and Information Systems: the Marriage of the Century? *undefined*, 2003.

Bradley Potteiger, Goncalo Martins, and Xenofon Koutsoukos. Software and attack centric integrated threat modeling for quantitative risk assessment. pages 99–108, 4 2016. . URL https://www.semanticscholar.org/paper/Software-and-attack-centric-integrated-threat-for-Potteiger-Martins/fdb3b4d599b111570f1c40b2b24c6c53e0dab091.

Sara Qamar, Zahid Anwar, Mohammad Ashiqur Rahman, Ehab Al-Shaer, and Bei Tseng Chu. Data-driven analytics for cyber-threat intelligence and information sharing. *Computers and Security*, 67:35–58, 6 2017. ISSN 01674048. .

M Rak, G Salzillo, and D Granata. ESSecA: An automated expert system for threat modelling and penetration testing for IoT ecosystems. *Computers and Electrical Engineering*, 99, 2022. ISSN 00457906. . URL https://www.sciencedirect.com/science/article/abs/pii/S0045790622000350.

Simona Ramanauskaite, Dmitrij Olifer, Nikolaj Goranin, and Antanas Čenys. Security ontology for adaptive mapping of security standards. *International Journal of Computers, Communications and Control*, 8(6):878–890, 2013. ISSN 18419844. .

Metwally Rashad, Ibrahim Reyad, and Mohamed Abdelfatah. Topic Modelling with Bag-of-concepts Document Representation. *NILES 2022 - 4th Novel Intelligent and Leading Emerging Sciences Conference, Proceedings*, pages 216–220, 2022. .

Karen Renaud and George R.S. Weir. Cybersecurity and the unbearability of uncertainty. In *Proceedings - 2016 Cybersecurity and Cyberforensics Conference, CCC 2016*, pages 137–143. Institute of Electrical and Electronics Engineers Inc., 10 2016. ISBN 9781509026579. .

Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, pages 44–51, 2009. ISBN 9780769537696. .

Alan Robinson and Andrei Voronkov. *Handbook of Automated Reasoning*. 2001. URL https://www.sciencedirect.com/book/9780444508133/handbook-of-automated-reasoning.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pages 399–408, 2 2015. . URL https://dl.acm.org/doi/10.1145/2684822.2685324.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C):53–65, 11 1987. ISSN 0377-0427. .

Jennifer Rowley and Richard Hartley. *Organizing Knowledge: An Introduction to Managing Access to Information*. 2000. URL http://eprints.rclis.org/12680/.

Clemens Sauerwein, Christian Sillaber, Andrea Mussmann, and Ruth Breu. Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. *Wirtschaftsinformatik 2017 Proceedings*, 1 2017. URL https://aisel.aisnet.org/wi2017/track08/paper/3.

Clemens Sauerwein, Irdin Pekaric, Michael Felderer, and Ruth Breu. An analysis and classification of public information security data sources used in research and practice. *Computers and Security*, 82:140–155, 5 2019. ISSN 01674048. .

Andreas Schaad and Dominik Binder. ML-supported identification and prioritization of threats in the ovvl threat modelling tool. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 12122 LNCS, pages 274–285. Springer, 6 2020. ISBN 9783030496685. . URL https://doi.org/10.1007/978-3-030-49669-2_16.

Andreas Schaad and Tobias Reski. "Open Weakness and vulnerability modeler" (OVVL): An updated approach to threat modeling. In *ICETE 2019 - Proceedings of*

the 16th International Joint Conference on e-Business and Telecommunications, volume 2, pages 417–424. SciTePress, 2019. ISBN 9789897583780. .

Christian A. Scholbeck, Henri Funk, and Giuseppe Casalicchio. Algorithm-Agnostic Interpretations for Clustering. 9 2022. URL https://arxiv.org/abs/2209.10578v1.

A. Sheth and C. Ramakrishnan. Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. *undefined*, 2003.

Adam Shostack. *Threat Modeling: Designing for Security*. Wiley, 4 2014.

Danny Velasco Silva and Glen Rodriguez Rafael. Ontologies for Network Security and Future Challenges. *Proceedings of the 12th International Conference on Cyber Warfare and Security, ICCWS 2017*, pages 541–547, 4 2017. URL http://arxiv.org/abs/1704.02441.

Rodrigo Silva and Fran Weidt Neiva. Systematic Literature Review in Computer Science - A Practical Guide. Technical report, Federal University of Juiz de Fora, 11 2016. URL https://www.researchgate.net/publication/320704338_Systematic_Literature_Review_in_Computer_Science_-_A_Practical_Guide.

Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics*, 5(2):51–53, 6 2007. ISSN 15708268. .

Jennifer Sleeman, Tim Finin, and Milton Halem. Understanding Cybersecurity Threat Trends Through Dynamic Topic Modeling. *Frontiers in Big Data*, 4, 6 2021. ISSN 2624909X. . URL /pmc/articles/PMC8275653//pmc/articles/PMC8275653/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC8275653/.

Peter Spyns, Yan Tang, and Robert Meersman. An ontology engineering methodology for DOGMA. *Applied Ontology*, 3(1-2):13–39, 2008. ISSN 18758533. .

Robert E. Stake. *The Art of Case Study Research*. Sage Publishing, 1995. URL https://us.sagepub.com/en-us/nam/the-art-of-case-study-research/book4954.

Stanford University. Protégé, 1999. URL https://protege.stanford.edu/.

Ryan Stillions. The DML model, 4 2014. URL http://ryanstillions.blogspot.com/2014/04/the-dml-model_21.html.

S. Subashini and V. Kavitha. A survey on security issues in service delivery models of cloud computing, 1 2011. ISSN 10848045.

J M Such, J Vidler, T Awais Seabrook, and A Rashid. Cyber Security Controls Effectiveness: A Qualitative Assessment of Cyber Essentials. Technical report, 2015.

M. Surridge, A. Chakravarthy, M. Hall-May, Xiaoyu Chen, B. Nasser, and R. Nossal. SERSCIS: Semantic Modelling of Dynamic, Multi-Stakeholder Systems. *undefined*, 2012.

Mike Surridge, Bassem Nasser, Xiayou Chen, Ajay Chakravarthy, and Panos Melas. Run-time risk management in adaptive ICT systems. In *Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013*, pages 102–110, 2013. ISBN 9780769550084. .

Hatma Suryotrisongko, Hari Ginardi, Henning Titi Ciptaningtyas, Saeed Dehqan, and Yasuo Musashi. Topic Modeling for Cyber Threat Intelligence (CTI). *2022 7th International Conference on Informatics and Computing, ICIC 2022*, 2022. .

Bill Swartout, R. Patil, K. Knight, and T. Russ. Toward Distributed Use of Large-Scale Ontologies t. *undefined*, 1997.

Frank Swiderski and Window Snyder. *Threat Modeling*. Microsoft Press, 2004. URL https://books.google.co.uk/books/about/Threat_Modeling.html?id=xawLAAAACAAJ&redir_esc=y.

Zareen Syed, A. Padia, Timothy W. Finin, M. L. Mathews, and A. Joshi. UCO: A Unified Cybersecurity Ontology. In *AAAI Workshop: Artificial Intelligence for Cyber Security*, 2016.

Samir Tartir, I. B. Arpinar, M. R. Moore, A. Sheth, and Boanerges Aleman-Meza. OntoQA: Metric-Based Ontology Quality Analysis. 2005.

G. R. Taylor. *Integrating Quantitative and Qualitative Methods in Research*. University Press of America, 2nd revised edition edition, 2005.

Steven Taylor, Michael Surridge, and Brian Pickering. Regulatory Compliance Modelling Using Risk Management Techniques. *SSRN Electronic Journal*, 10 2020. ISSN 1556-5068. . URL https://www.ssrn.com/abstract=3716778.

Chun Man Tsang, Tom Bell, Antonios Gouglidis, and Mo El-Haj. Deciphering Cyber Threats: A Unifying Framework with GPT-3.5, BERTopic and Feature Importance. In *The 1st International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security (NLPAICS), Lancaster, UK. 2024.*, pages 175–185, 2024.

U.S Bureau of Labor Statistics. Distribution of private sector firms by size class, not seasonally adjusted, 2019. URL https://www.bls.gov/web/cewbd/table_g.txt.

Michael Uschold and Michael Grüninger. Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, 11(2), 1 1996. URL https://www.researchgate.net/publication/302937543_Ontologies_Principles_methods_and_applications.

Muhammad Usman, Ricardo Britto, Jürgen Börstler, and Emilia Mendes. Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method, 5 2017. ISSN 09505849.

Thomas D. Wagner, Khaled Mahbub, Esther Palomar, and A. E. Abdallah. Cyber threat intelligence sharing: Survey and research directions. *Computers and Security*, 87: 101589, 11 2019. ISSN 01674048. .

Ju An Wang and Minzhe Guo. *OVM: An Ontology for Vulnerability Management*. 2009. ISBN 9781605585185.

Wenjun Xiong and Robert Lagerström. Threat modeling – A systematic literature review. *Computers & Security*, 84:53–69, 7 2019. ISSN 0167-4048. .

Y Xiong, Z Yang, B Wang, P Xun, and T Deng. False sequential command attack of large-scale cyber-physical systems. *Electronics (Switzerland)*, 7(9), 2018. ISSN 20799292. .

Robert K. Yin. *Case Study Research and Applications*. Sage Publishing, 6th edition edition, 2017. URL https://us.sagepub.com/en-us/nam/case-study-research-and-applications/book250150.

Ana Sofía Zalazar, Luciana C. Ballejos, and S. Rodriguez. Security and Compliance Ontology for Cloud Service Agreements. *undefined*, 2017.

He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. Topic Modelling Meets Deep Neural Networks: A Survey. *IJCAI International Joint Conference on Artificial Intelligence*, pages 4713–4720, 2 2021. ISSN 10450823. . URL https://arxiv.org/abs/2103.00498v1.