Bournemouth University

FACULTY OF SCIENCE & TECHNOLOGY

MSc Internet of Things with Cyber Security

May 2022

APT Groups & Malware Attribution.

by

Paul-David Jarvis

Faculty of Science & Technology

Department of Computing and Informatics

Final Year Project

Abstract

The purpose of this project is to provide a look into APT groups and the methods that they employ in their campaigns against their victims. We must have a solid grasp of these tactics and techniques that these APT groups use so we could begin to devise countermeasures to make organisations and people safer online. We collected our data from three reliable sources on the data analysis platform Kaggle and used deductive reasoning to produce our findings. We found that a majority of our APT groups used the exact YARA rules suggesting a similar attack method thus a link between them. We also found that Window's executable files were the most common method of attack with our groups. We also revealed that our top common ATT&CK techniques could be associated with stages on cyber kill chains. These findings by themselves are interesting but to use this data effectively, we need to analyse it using metrics that we discuss later on. We know that the standard for causing APT groups the most pain is Tactics, Techniques, and Procedures which we can get from the YARA rules and ATT&CK techniques. Knowing that the common attack method is Window's executables is not surprising but more interesting are the methods that we rarely saw such as an Android application; if we are aware that an APT group uses a niche method and know their methods and whom they target we might be able to predict whom they will attack based on this intelligence.

Dissertation Declaration

I agree that, should the University wish to retain it for reference purposes, a copy of my dissertation may be held by Bournemouth University normally for a period of 3 academic years. I understand that once the retention period has expired my dissertation will be destroyed.

Confidentiality

I confirm that this dissertation does not contain information of a commercial or confidential nature or include personal information other than that which would normally be in the public domain unless the relevant permissions have been obtained. In particular any information which identifies a particular individual's religious or political beliefs, information relating to their health, ethnicity, criminal history or sex life has been anonymised unless permission has been granted for its publication from the person to whom it relates.

Copyright

The copyright for this dissertation remains with me.

Requests for Information

I agree that this dissertation may be made available as the result of a request for information under the Freedom of Information Act.

p.jarvis

Signed: _____

Name: Paul-David Jarvis

Date: 01/05/2021

Programme: BSc (Hons) Forensic Computing and Security

Original Work Declaration

This dissertation and the project that it is based on are my own work, except where stated, in accordance with University regulations.

p.jarvis

Signed: _____

Name: Paul-David Jarvis

Date: 01/05/2021

Acknowledgements

Firstly, I'd like to acknowledge my supervisor, Prof. Katos, all his help, guidance and motivation as made this project extremely fun to work on. Secondly, I'd like to acknowledge VirusTotal, It was early in this project that we found that it wouldn't be advisable to use VirusTotal's API without their premium key that would allow us to make unlimited requests to retrieve data on our SHA256 hashes. Discussing my project with them, they were happy to upgrade my account and provide me with an academic API key. Thirdly I'd like to give my gratitude to smelley vx, the founder of VX-Underground that was happy enough to provide me access to VX-Underground's private database and their API key. Fourthly I would like to acknowledge Prof. Patsakis from University of Piraeus, his advice and help has elevated the data analysis performed in this project. I would be thrilled to work with any of the aforementioned people in the future.

Contents

Abstract ii

Acknowledgements v

1 Introduction and Motivation 2

1.1 Aims and objectives . . . 3

1.2 Research Questions . . . 3

1.3 Success criteria for each objective . . . 4

1.4 Methodological Approach . . . 4

1.5 Overview of the remaining chapters . . . 4

2 Literature Review 6

2.1 APT Groups . . . 6

2.2 Cyber Situational Awareness . . . 6

2.3 Pyramid of Pain . . . 7

2.4 Structured Threat Information Expression . . . 8

2.5 Cyber Threat Intelligence . . . 9

2.5.1 Indicators of Compromise . . . 10

2.5.2 Tactics, Techniques & Procedures . . . 11

2.5.3 Effective Threat Intelligence . . . 11

2.5.4 Cyber Kill Chains . . . 12

2.6 Dataset . . . 14

2.6.1 VX-Underground . . . 14

2.6.2 VirusTotal . . . 14

2.6.3 Triage . . . 15

2.6.4 Kaggle . . . 15

3 Methodology 16

3.1 Data Gathering . . . 16

3.1.1 VX-Underground . . . 16

3.1.2 VirusTotal . . . 17

3.1.3 Triage . . . 18
3.2 Data Sets . . . 19
3.2.1 Evaluation against the NIST Criteria . . . 20
3.3 Kaggle . . . 21

4 Analysis and Findings **22**
4.0.1 Group's Format Speciality . . . 22
4.0.2 Suggested Threat Label . . . 24
4.0.3 MITRE ATT&CK . . . 26
4.0.4 YARA Rules . . . 28
4.0.5 Malware Families . . . 32

5 Conclusion **34**
5.1 Research Questions . . . 34
5.1.1 Research Question 1 . . . 34
5.1.2 Research Question 2 . . . 34
5.1.3 Research Question 3 . . . 34
5.1.4 Research Question 4 . . . 35
5.2 Future Work . . . 35
5.3 Limitations & Caveats . . . 36

Bibliography **40**

Appendix A Project Proposal **41**

Appendix B Ethics Checklist **45**

Appendix C Artifacts **47**
C.0.1 Kaggle Notebook . . . 47
C.0.2 APT Analysis Datasets . . . 47

Appendix D Scripts **49**
D.0.1 VirusTotal . . . 49
D.0.2 VX-Underground . . . 49
D.0.3 Triage . . . 50
D.0.4 Linux . . . 50
D.0.5 YARA Rules . . . 51

List of Figures

1 CSA Concept. (MITRE (1997)) 7
2 The Pyramid of Pain. (Herring (n.d.)) 8
3 APT1 STIX Diagram. (STIX (2017)) 9
4 APT1 STIX Diagram Legend. (STIX (2017)) 9
5 The Lockheed Kill Chain. (Hutchins et al. (2011)) 13
6 The Unified Kill Chain. (Pols and van den Berg (2017)) 14

7 Datasets & schemas 20

8 Percentage of File Types. 23
9 Top 10 Common File Types Occurrence. 24
10 Top 10 Suggested Threat Label Occurrences. 26
11 Top 10 ATT&CK Techniques Occurrences. 27
12 Top 10 YARA Rules Occurrences. 29
13 Interesting YARA Rules. 30
14 APT Attribution and Linkage. 32
15 Top 10 Malware Families Occurrences. 33

16 First Page of Project Proposal. 41
17 Second Page of Project Proposal. 42
18 Third Page of Project Proposal. 43
19 Fourth Page of Project Proposal. 44

20 First Page of Ethics Checklist. 45
21 Second Page of Ethics Checklist. 46

List of Tables

1 Table of Abbreviations 1

4.1 File Types Count. 24
4.2 Suggested Threat Label Count. 26
4.3 ATT&CK Techniques Count. 28
4.4 Cosine Similarity of YARA Rules & APT Groups. 31

Abbreviation	Meaning
APT	Advanced Persistent Threat
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
APK	Android Package Kit
API	Application Programming Interface
CLI	Command Line Interface
C2	Command-and-Control
CSV	Comma-Separated Values
CKC	Cyber Kill Chains
CSA	Cyber Situational Awareness
CTI	Cyber Threat Intelligence
IoCs	Indicators of Compromise
JSON	JavaScript Object Notation
LKC	Lockheed Kill Chain
MO	Modus Operandi
NHS	National Health Service
NIST	National Institute of Standards and Technology
PoP	Pyramid of Pain
RAT	Remote Access Trojan
SOC	Security Operation Center
STIX	Structured Threat Information Expression
TTPs	Tactics, Techniques & Procedures
UKC	Unified Kill Chain
VBS	Visual Basic Script

Table 1: Table of Abbreviations

1 INTRODUCTION AND MOTIVATION

The continuous evolution and digitisation that is happening in the world have seen a constant increase in Internet of Things (IoT) devices and more dependency on technology has seen the undesirable and detrimental effect that is cyber threat actors with new tools and extended capabilities transforming the threat landscape. Hence the need for cyber security. This refers to the measures taken to ensure and protect the confidentiality, integrity, and availability of software and hardware, as well as information. This includes Personally identifiable information stored in databases to Programmable Logic Controllers and other Industrial Control Systems.

Introducing Advanced Persistent Threat (APT) groups that possess the means and abilities to initiate campaigns and long-term and purposeful attacks against a specific target, which has the characteristics of complex attack means, long duration, and great harmfulness (Liu et al. (2019)). The complexity and resources that these attacks exhibit usually means that they're nation-state backed and target rival governments. APT groups can gain an initial foothold into their target's systems and could go years without being detected while siphoning off information and forwarding it back to a Command-and-Control (C2) server in their control. This is why these groups are considered the most dangerous because their attacks have the capability of causing the most disruption and damage.

Sharing cyber threats and related information is vital because of this. Raising awareness for previously discovered threats could involve solutions for the next attacks. Knowing this knowledge and other information out there regarding an APT group's Tactics, Techniques & Procedures (TTPs) could disrupt any future campaigns and attacks.

Throughout this project, we have collected an incredible amount of data on campaigns that have already happened and contextualised it all around the golden standard of models used when evaluating the quality of Cyber Threat Intelligence (CTI). The Pyramid of Pain (PoP), the Unified Kill Chain (UKC), the Lockheed Kill Chain (LKC) and the characteristics that make CTI effective as stated by the National Institute of Standards and Technology (NIST). The mapping of these models allows us and other end users to gain a better understanding of the CTI that is being shared. Consider the low-level Indicators of Compromise (IoCs) hash which alone might offer a timely identification of malware but nothing else. Whereas, high-level IoCs like TTPs provide us more with how capable the threat actor was. In addition, mapping the Cyber Kill Chains (CKC) to

the TTPs exhibited demonstrates the stages of attacks thus providing information on the Modus Operandi (MO) of an APT group and potential countermeasures.

An example is the infamous Lazarus Group, sponsored by the Reconnaissance General Bureau, a military and intelligence arm of North Korea (Jasper (2019)). Suspected to be behind the notorious WannaCry incident in May 2017. The self-propagating ransomware spread through internal networks and over the public internet by exploiting a vulnerability in Microsoft's Server Message Block protocol (Satheesh Kumar et al. (2018)). The piece of malware hit 230,000 computers globally with thousands of National Health Service (NHS) hospitals and surgeries in the UK being targeted leaving ambulances to be rerouted which left people in need of urgent care. Estimated to cost the NHS £92 million with over 19,000 appointments having to be cancelled. The United Kingdom wasn't the only country to be affected with computer systems in 150 countries being crippled causing an estimated amount of 4 billion dollars of losses (Rogers and Kanth (2021)).

1.1 Aims and objectives

This project aims to conduct an empirical evaluation of APT groups and the methods that they employ. As such, this project came with a series of technical challenges which were mapped to the following project objectives.

- Collect a sufficient amount of data to perform statistical analysis.
- Develop the necessary scripts that scrape the needed data using VirusTotal, VX-Underground and Triage's Application Programming Interface (API) to automate the collection of the data we need.
- Develop a series of Kaggle notebooks that will manipulate the data we collect to confirm or refute our hypotheses.

The methodology follows a two-phased approach. First, we conduct an initial exploratory phase where we get a feel of the domain. Secondly, quantitative analysis will be conducted where correlations and associations will be made between APT groups, hashes and other attributions.

1.2 Research Questions

RQ_1 Is it feasible to use low-value IoCs such as SHA256 hashes that are in abundance to construct the CKC?

RQ_2 Are malware signatures sufficient IoCs to perform APT attribution?

RQ_3 Can we provide efficient CTI by using sparse data?

RQ_4 Is it possible to predict potential targets based on intelligence regarding APTs and their TTPs?

1.3 Success criteria for each objective

- The scripts that were written in bash using APIs from VirusTotal, VX-Underground and Triage must compile correctly and return the necessary and accurate data. This data includes YARA rules, C2 servers and any other information that provides a detailed look into APT groups.
- The aforementioned scripts must return the necessary data and the output must be converted properly so Python's Panda module can read it.
- We have to create Kaggle notebooks with a satisfying amount of python code dedicated to statistical analysis and produce valuable results, plots, and graphs that provide insight into APT groups.

1.4 Methodological Approach

In Chapter 3, we go into more depth about the methodology approach that we took throughout this project. We first started by collecting our list of SHA256 samples, this was whittled down to include necessary hashes. They were then used as input for our VirusTotal, VX-Underground and Triage scripts to request and store information on the specific hash. Once this data was stored and converted, they were uploaded as data sets to Kaggle where we began writing python code that would manipulate and analyse our data. The software library pandas were used to store our data into data frames. Plotly was used to create charts and plots used to display our data.

A dataset diagram was created to show how a specific piece of data would be connected to another. E.g. we used YARA rules as representations to search data. We combined APT groups with our data on IP addresses and domains, families and Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) data to represent how all these can be combined and how they work together to show how APT groups operate. In addition, we performed a cosine similarity test to identify collaboration between two APT groups based on how similar the TTPs are.

1.5 Overview of the remaining chapters

In Chapter 2 we set out to review, study and evaluate any already available literature that could be relevant to this project. This included the origins of APT groups and the standards that should be considered when dealing with CTI; we looked at the relevant CKCs, the PoP and the relevant

IoCs and the characteristics that make CTI effective as drafted by the NIST. Chapter 3 is where a more in-depth methodological approach can be found. We explain our proposed approach to the collection and analysis of our data. Chapter 4 contains the final analysis and discussion of the findings including deductive reasoning. Finally, in chapter 5 you can find our conclusion where we summarise the main findings and discuss the potential for any future research and the challenges we faced.

2 LITERATURE REVIEW

2.1 APT Groups

The term APT Attack was first proposed by United States Air Forces in 2006. According to ”SP800-53 management information security risk” published by the US NIST in 2013, in the definition of APT: proficient technology attackers use a variety of attack programs (network, physical and fraud) with rich resources to achieve the purpose of the attack. In recent years, the security situation in cyberspace has become more and more serious, and the advanced threat has become the most threatening cyber attack. Almost all important industries, such as government, finance, electric power and education, are threatened by APT attacks (Zhang et al. (2017)).

APT is a military term adapted into the information security context that refers to attacks carried out by nation-states. APT is defined by the combination of three words (Alshamrani et al. (2019)). Which are:

- **Advanced:** APT attackers are usually well-funded with access to advanced tools and methods required to perform an APT attack. These advanced methods include the use of multiple attack vectors to launch as well as to keep the attack going.
- **Persistent:** APT attackers are highly determined and persistent and they do not give up. Once they get into the system, they try to stay in the system for as long as they can. They plan the use of several evasive techniques to elude detection by their target's intrusion detection systems. They follow the “low and slow” approach to increase the rate of their success.
- **Threat:** The threat in APT attacks is usually sensitive data loss or impediment of critical components or mission. These are rising threats to many national entities and organizations that have advanced protection systems guarding their missions and/or data.

2.2 Cyber Situational Awareness

The growing dependency on cyberspace has greatly increased the need for situation awareness. Essentially, we need to understand the environment so we can predict and respond to any cyber problems that might occur. Significant risks to individuals, organisations and potentially national security have been caused by vulnerabilities in networks and systems. MITRE has released a

comprehensive Cyber Situational Awareness (CSA) concept that explains the required investment in data collection, data management and analysis to maintain an ongoing picture of how the computer systems, networks, and users are operating in an organization (MITRE (1997)).

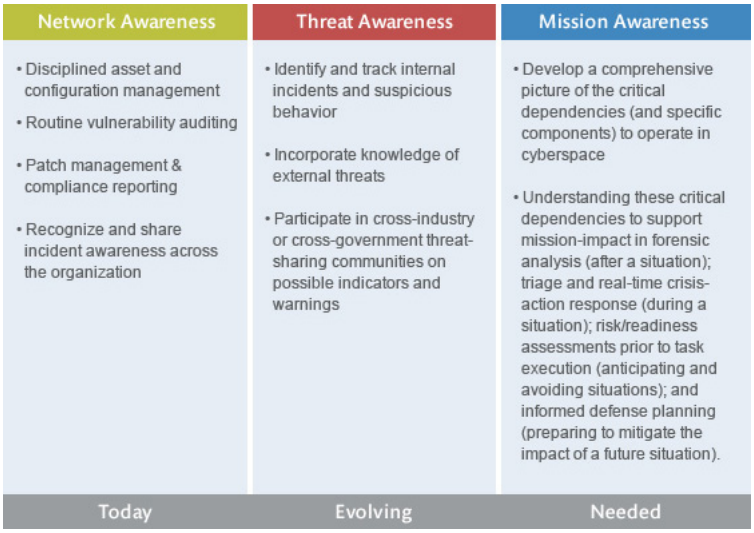


Figure 1: CSA Concept. (MITRE (1997))

2.3 Pyramid of Pain

In 2013, David J Bianco introduced a concept known as 'The Pyramid of Pain' illustrated in Figure 2. The PoP identifies the different types of indicators that can be used to identify a Threat Actor's activity and the pain it will cause a Threat Actor if one denies them the ability to use those indicators. The PoP colour codes the Hash, IP Addresses and Domain Names as Green implying that it is extremely easy for a Threat Actor to change these indicators. The host and network artefacts are colour coded as light yellow indicating that it is annoying for a Threat Actor to change the host and network artefacts. The Tools are colour coded as yellow and termed very painful for a Threat Actor to change. A Threat Actor needs to identify a new tool with the same capability, and spend considerable time in development and training. The TTPs are colour-coded red and deemed most painful for a Threat Actor to change. This forces a Threat Actor to learn new behaviours which is the hardest thing to do (Warikoo (2021)).

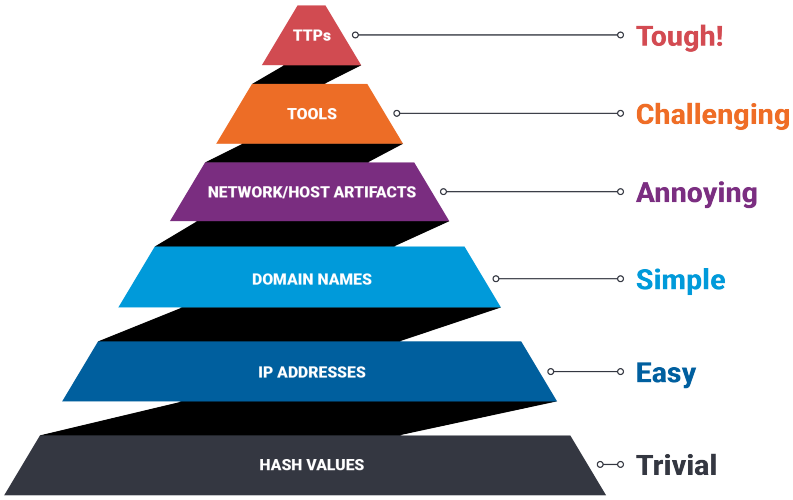


Figure 2: The Pyramid of Pain. (Herring (n.d.))

2.4 Structured Threat Information Expression

Structured Threat Information Expression (STIX) was introduced by MITRE and supported by the US government through discussions among security operations and CTI experts (Okada et al. (2021)). Contributing and ingesting CTI becomes a lot easier. With STIX, all aspects of suspicion, compromise and attribution can be represented clearly with objects and descriptive relationships. STIX information can be visually represented for an analyst or stored as JavaScript Object Notation (JSON) to be quickly machine readable. STIX's openness allows for integration into existing tools and products or utilisation for your specific analyst or network needs (STIX (2017)). STIX has converted Mandiant's APT1 report into JSON representation that allows us to view the report using their language and represents all attributes such as the threat actors, identity of the hackers and IoCs are shown in figure 3.

Figure 4 shows us the legend automatically created that tells us what each symbol means. We can see from figure 3 that the threat actor Ugly Gorilla, has been attributed to the identity of Wang Dong and JackWang which has been discovered to be Ugly Gorilla's real name and an alias. We see that Wang Dong uses MANITSME and WEBC2-UGX. We also see that the domain hugesoft.org and arrowservice.net are both indicators associated with Ugly Gorilla.

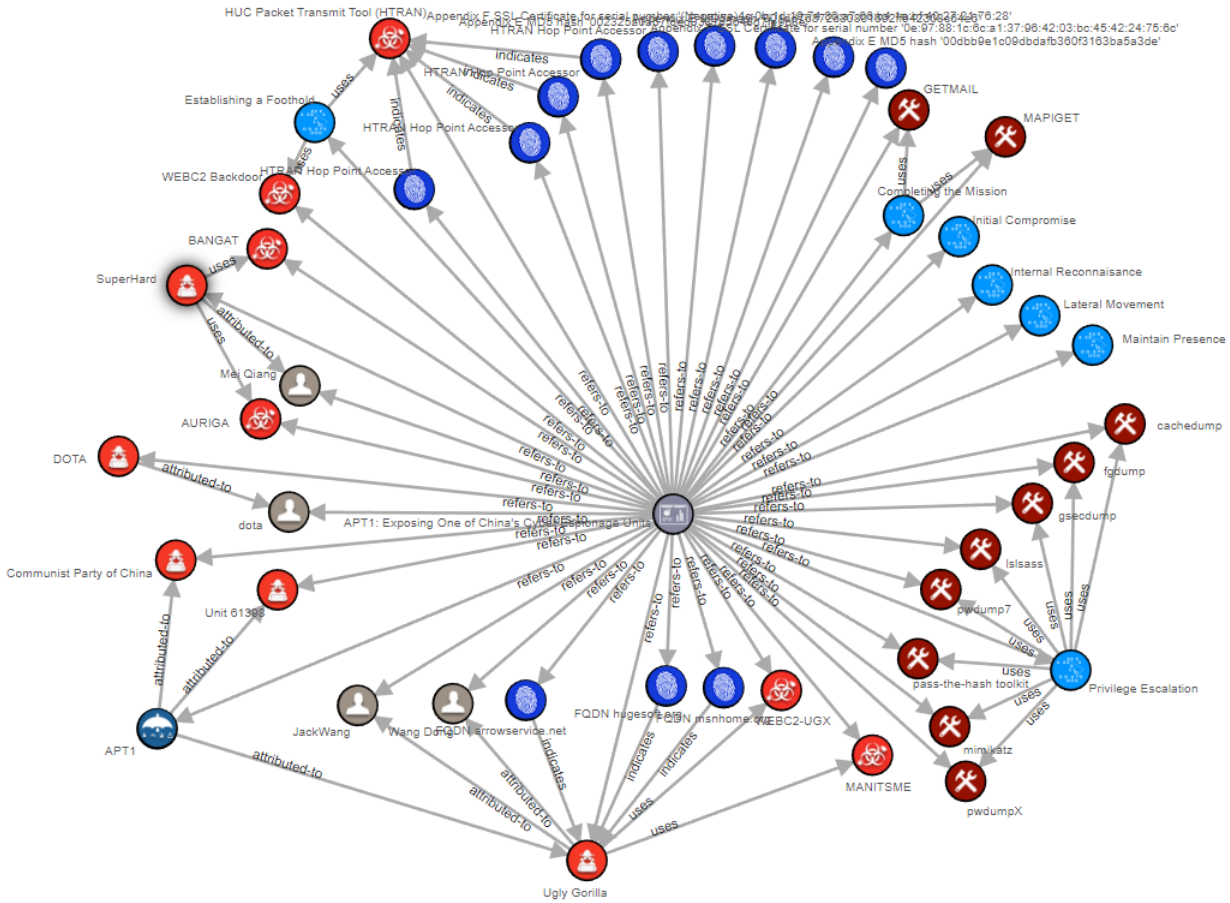


Figure 3: APT1 STIX Diagram. (STIX (2017))

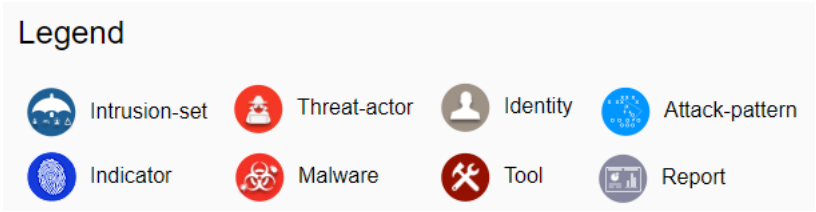


Figure 4: APT1 STIX Diagram Legend. (STIX (2017))

2.5 Cyber Threat Intelligence

Threat intelligence is data that is collected, processed, and analysed to understand a threat actor's motives, targets, and attack behaviours. Threat intelligence enables us to make faster, more informed, data-backed security decisions and change their behaviour from reactive to proactive in the fight against threat actors. The threat intelligence that we have collected is important as we hope it will provide useful insight into the APT group's motives and tactics, TTPs. The main

challenge with this, however, is that threats are constantly evolving but using threat intelligence lifecycles provides a framework to enable teams to optimize their resources and effectively respond to the modern threat landscape (Baker (2022)). CrowdStrike's life cycle is as follows:

- **Requirements:** Set roadmap for a threat intelligence operation.
- **Collection:** Collect information to satisfy objectives.
- **Processing:** Raw data is processed into a format suitable for analysis.
- **Analysis:** Finding answers to questions posed during the requirements phase.
- **Dissemination:** Translating analysis into a digestible format for stakeholders.
- **Feedback:** Receive stakeholder feedback and establish adjustments for the future.

Throughout this project we have followed a similar framework; The requirements stage is where the discussion was had regarding what and where data was collected and the questions posed about what this data could help with. The collection stage was where we executed all the scripts that we wrote that used APIs and scrapped the data. Processing happened when we converted the JSON formatted data to Comma-Separated Values (CSV) format. The analysis stage was using Kaggle to combine and analyse our datasets. The dissemination stage happened when all the data was analysed and discussed what it shows and feedback happened when we submitted this project and if we open our the research collected to other stakeholders like Security Operation Center (SOC) analysts.

2.5.1 Indicators of Compromise

IoCs are forensic artefacts that are used as signs that a system has been compromised by an attack or that a system has been infected with particular malicious software, such as domains/IPs of APT, MD5 hashes of attack files, virus signatures (Zhang et al. (2019)). Further along in this thesis, we will introduce the data we collected from the several reliable sources that we have mentioned and touched upon below where we will be analysing IoCs that have been discovered to be associated with our list of SHA256 hashes in an attempt to provide any useful CTI. Such as any domains and IP addresses that could have potentially been used as C2 servers for these attacks and attempt to link them to one or more APT groups and potentially uncover their MO when it comes to whether they work alone or with other APT groups. We have also collected ssdeep, tlsh, vhash and more as signatures that we might be able to group and classify.

2.5.2 Tactics, Techniques & Procedures

A publication from NIST labelled ”Guide to Cyber Threat Information Sharing” describes TTPs as the behaviour of a threat actor. Tactics are high-level descriptions of behaviour, techniques are detailed descriptions of behaviour in the context of a tactic, and procedures are even lower-level, highly detailed descriptions in the context of a technique. TTPs could describe an actor's tendency to use a specific malware variant, order of operations, attack tool, delivery mechanism (e.g., phishing or watering hole attack), or exploit (Johnson et al. (2016)).

The MITRE ATT&CK is a public, knowledge-based repository of adversary tactics and techniques created by the MITRE Corporation to create a standardized taxonomy for the sharing of malware information. The repository structure reflects each step of an adversary's attack lifecycle, and the Tactics and Techniques model provides a common terminology for both the defensive and offensive sides of cyber security. Furthermore, the MITRE ATT&CK standard allows researchers and professionals to understand and analyze how a specific adversarial behaviour is used to achieve a goal and better respond to it (Chierzi and Merces (2021)).

2.5.3 Effective Threat Intelligence

The aforementioned publication by NIST (Johnson et al. (2016)) outlines the following criteria as a method of showing what characteristics that threat intelligence exhibits.

- **Timely:** Intelligence should be delivered rapidly, allowing a sufficient opportunity for the recipient to anticipate the threat and respond.
- **Relevant:** Intelligence should be relevant to the recipient; attacks their organisation is likely to face and adversaries they're likely to encounter.
- **Accurate:** Intelligence should be correct, complete, and unambiguous.
- **Specific:** Intelligence should depict the incident or adversary in a way that recipients can understand how the threat may affect them and allows them to evaluate the next course of action.
- **Actionable:** Intelligence should identify actions that the recipient can take to counter the threat or provide sufficient information that allows them to develop a suitable response.

The reason why it is important to discuss these characteristics is that we can use them to evaluate the data that we have collected. For example, our data includes information on MITRE's ATT&CK that tells us the tactics and techniques that an APT group uses; this information is considered relevant as it addresses the likely threats an attacker would employ as well as provides a look at

the type of adversaries that would attack the organisation.

The information, in this case, was collected using Triage's API, there information was collected from other researchers in the field that have encountered a specific malware and associated using MITRE's ATT&CK website which suggests that this data came from a credible and accurate source.

TTPs are very straightforward and specific. The hash "e8ddefd237646a47debc01df9aa02fbcae40686f96b7860511c73798c7546201" has been associated with the technique "Access Token Manipulation" which tells us that this malware modifies access tokens to bypass access controls.

The information we have is extremely actionable; especially the aforementioned example. We are aware that the above hash manipulates access tokens, fortunately for us ATT&CK also provides mitigation methods, the recipient can either use "Privileged Account Management" or "User Account Management". We are aware of these countermeasures because we are aware of the relevant and accurate data.

2.5.4 Cyber Kill Chains

Lockheed Kill Chain

The LKC developed by Lockheed Martin Computer Incident Response Team is an attack model, which defines an attack as a chain of action. It is a structured attack since the attacker progresses the attack in an ordered chain according to the plan. The Kill Chain technique is described by the US Department of Defence to attack a target, where they have defined the Kill Chain with some stages such as, find, fix, track, target, engage and assess. The Kill Chain has been applied in other areas including cyber security. In cyber security, it is used to describe some attack steps within a countermeasure framework (Al-Mohannadi et al. (2016)). The LKC stages are seen in figure 5.

Figure 5: The Lockheed Kill Chain. (Hutchins et al. (2011))

Unified Kill Chain

The second CKC we have researched and will be using to see if we can compare the stages with MITRE ATT&CK techniques is the UKC developed by Paul Pols to offer researchers significant improvements over the more malware-focused CKC developed by Lockheed. the UKC provides insights into the ordered arrangement of phases in attacks from their beginning to their completion by uniting and extending existing models. Other improvements that the UKC include: explicating the role of users by modelling social engineering, recognizing the crucial role of choke points in attacks by modelling pivoting, covering the compromise of integrity and availability in addition to confidentiality and elucidating the overarching objectives of threat actors (Pols and van den Berg (2017)). The 18 tactics that the UKC identified are seen in Figure 6.

The Unified Kill Chain		
1	Reconnaissance	Researching, identifying and selecting targets using active or passive reconnaissance.
2	Weaponization	Preparatory activities aimed at setting up the infrastructure required for the attack.
3	Delivery	Techniques resulting in the transmission of a weaponized object to the targeted environment.
4	Social Engineering	Techniques aimed at the manipulation of people to perform unsafe actions.
5	Exploitation	Techniques to exploit vulnerabilities in systems that may, amongst others, result in code execution.
6	Persistence	Any access, action or change to a system that gives an attacker persistent presence on the system.
7	Defense Evasion	Techniques an attacker may specifically use for evading detection or avoiding other defenses.
8	Command & Control	Techniques that allow attackers to communicate with controlled systems within a target network.
9	Pivoting	Tunneling traffic through a controlled system to other systems that are not directly accessible.
10	Discovery	Techniques that allow an attacker to gain knowledge about a system and its network environment.
11	Privilege Escalation	The result of techniques that provide an attacker with higher permissions on a system or network.
12	Execution	Techniques that result in execution of attacker-controlled code on a local or remote system.
13	Credential Access	Techniques resulting in the access of, or control over, system, service or domain credentials.
14	Lateral Movement	Techniques that enable an adversary to horizontally access and control other remote systems.
15	Collection	Techniques used to identify and gather data from a target network prior to exfiltration.
16	Exfiltration	Techniques that result or aid in an attacker removing data from a target network.
17	Impact	Techniques aimed at manipulating, interrupting or destroying the target system or data.
18	Objectives	Socio-technical objectives of an attack that are intended to achieve a strategic goal.

Figure 6: The Unified Kill Chain. (Pols and van den Berg (2017))

2.6 Dataset

2.6.1 VX-Underground

VX-Underground was officially made in May 2019 by smelly_vx as a successor to vxHeaven, a computer virus library. An invaluable source of malware samples that were the foundation of this project. Considered a very controversial group with been banned from a lot of web hosts as they were not wanting to host a malware sample library. In an interview with smelly_vx and Dmitry Smilyanets, product manager from Recorded Future analyst put on record that VX-Underground hosts the largest collection of malware source code, samples and papers on the internet with about 35 million samples overall and their internal group is composed of people from anti-virus companies, threat intelligence companies, Incident response companies, Universities (academics) and “Red Team” companies (Smilyanets (2022)).

2.6.2 VirusTotal

VirusTotal is a free service that provides a place of collaboration between members of the antivirus industry, researchers and end-users of all kinds. Founded in 2004, VirusTotal allows end-users to analyse files and URLs for viruses, worms, trojans and other kinds of malicious content. Fortune 500 companies, governments and leading security companies are all part of the VirusTotal community, which has grown to over 500,000 registered users (Virustotal (2004)).

2.6.3 Triage

Hatching Triage is a state-of-the-art malware analysis sandbox designed for cross-platform support (Windows, Android, Linux, and macOS), high-volume malware analysis capabilities, and malware configuration extraction for dozens of malware families (Hatching (2018)). Triage was suggested to us by Associate Professor Konstantinos from the University of Piraeus and provides us with some incredible information and analysis on our hashes.

2.6.4 Kaggle

Kaggle is an online community that includes everyone involved in big data from data scientists and Machine Learning practitioners to academic researchers such as myself. Kaggle allowed us to upload our data sets and gives us a brilliant web-based python environment where we are able to analyse our data with the ability to share the online environment and datasets with other data scientists and researchers which is why we chose it as our data-science environment to ensure the most efficient way of collaboration between the people working on this project.

3 METHODOLOGY

3.1 Data Gathering

3.1.1 VX-Underground

This project began using a list of hashes associated with APT groups from VX-Underground's database. Communicating with smelly_vx; the founder of VX-Underground, we were allowed access to VX-Underground's online database and access to their API. From there we developed two scripts written in bash that utilise their Command Line Interface (CLI) and another that utilises their online API, Seen in D.0.2. These scripts allowed us to retrieve and store the information that they had on a specific hash. Information such as C2 servers that these hashes communicated with and whether any YARA rules were associated with them. We can later enrich this information by identifying the origin of the C2 servers to analyse whether more than one APT group are communicating with the same C2 servers which can suggest that there is potentially teamwork and collaboration between the APT groups.

The scripts we wrote stored the data we requested in JSON format then we turned it into CSV format. From there we identified and eliminated any data that wasn't necessary. The data we kept were:

- **Botnet:** The botnet data was hard to interept; the information included "Adrell" and "Anjola" which we assume are the names of botnets associated with these hashes, however, there is no clear definition of what these names represent which makes drawing any conclusions difficult.
- **C2 Servers:** The C2 data is considered one of the most useful pieces of data alongside YARA rules that we retrieved from VX-Underground. It is logical to assume that nation-state APT groups exfiltrate the data that they have stolen to C2 servers originating from their region to ensure the most protection in regards to the legality of being able to search other countries servers. The presumption could be made that the APT group responsible and wherever the C2 server that they're using could be of the same nationality.
- **Family:** Family was another interesting tag associated with these hashes that unfortunately were less rich than we hoped. 41% of the 10,402 rows we had couldn't be identified to

belong to any family. That being said, we still had some interesting data from the 59%; we can link an APT group to a hash and that hash to a family of malware if applicable. E.g. we can tell that APT 29, also known as NOBELIUM uses Cobalt Strike.

- **Rule:** Rule's data by themselves fail to provide any additional information that isn't provided by the Family. However, we never disregarded this association as it can be used to enrich the Family.
- **Tag:** Undeniably the most useful association that we were provided with. This encompasses some of the aforementioned pieces of data but more importantly, it can provide us with YARA rules associated with the hashes. For example, the SHA256 hash "0007ccdddb12491e14c64317f314c15e0628c666b619b10aed199eefcfe09705" has the YARA rule "win_registry" associated to it, researching that rule tells us that malware found to have this rule affects computer's registries.

3.1.2 VirusTotal

VirusTotal was another malware engine that we made sure to utilise as this provided a lot of different information regarding our hashes. The main limitation with VirusTotal was their API; anybody can gain access to a public API key, however, for this project a public API key was not advisable due to the constraints and restrictions that follow, mainly the descriptions of the number of requests and request rate being 500 per day and 4 per minute. Taking into consideration that we were analysing roughly 35,000 hashes and the nature of this project being research and we granted a premium API key from VirusTotal that allowed us to request 20,000 per day. We created a bash script that would read our list of hashes then request all information stored on them and save them, seen in D.0.1.

We retrieve plenty of data on each hash that will help us plot APT groups and their methods. Basic properties such as the other types of hashes: TLSH, MD5, SSDEEP and Vhash were retrieved as well as historical data that will be used to plot timelines so we can conclude when a specific malware was first encountered to the time it took for them to be submitted. The list of data saved is:

- **Vhash, TLSH, SSDEEP:** The different types of hashes associated with the malware. Most of the hashes do not provide any additional information about an APT group or the malware they used. However, we might be able to SSDEEP and TLSH to search for any tangible intersections between the APT groups and start to cluster.

- **Type Tag, Description, Extension, Magic:** These tags can provide us with a good idea of what type of malware is being used by APT groups. For example, we can see that SHA256 ”afcbaae700e1779d3e0abe52bf0f085945fc9b6935f7105706b1ab4a823f565f” has the type description of Win32 EXE and we can then link that hash to the APT group Primitive Bear; which the Ukrainian government associated it to Russia’s Federal Security Service Center 18.
- **First Submission Date, Last Submission Date, First Seen In The Wild, Last Analysis Date, Last Modification Date:** Using this historical data we could potentially plot out the activity of a specific APT group and malware. We could potentially see whether APT groups are more active at a certain point or whether any APT group’s activities overlap which could again suggest collaboration between groups. We can also get a good idea of how stealthy an APT group is by looking at when a specific malware was first encountered until it was first submitted.
- **Suggested Threat Label** VirusTotal provides us with the most common label associated with malware stated by an extensive list of malware vendors such as Malwarebytes, Kaspersky, CrowdStrike and many more. This information is extremely interesting as it can provide a well-researched guess into what the malware is. For Example, SHA256 ”021858a878b8cb20b031817229e25b07daffd43be259df7f1bebde694a84f84d” has the suggested threat label assigned to it ”trojan.shadowpad/razy” which tells us that the above hash is a version of the Remote Access Trojan (RAT) ShadowPad; ShadowPad is a network intrusion malware affiliated to both the Chinese Ministry of State Security and the People’s Liberation Army. ShadowPad is depicted as a “back-door ‘Trojan’ malware which creates a secret path from a targeted system to a command and control server to extract information (Narayanan (2021)).

3.1.3 Triage

Hatching Triage was the second malware analysis sandbox that we used to provide any additional threat intelligence on our list of samples. When creating an account and providing them with my academic email and information on this project we were upgraded to the ”researcher” account role. Interestingly we faced the most challenges when attempting to use their API and convert the malware data from JSON style to CSV.

The data that we kept from the scripts were:

- **Botnet:** The botnet dataset contains a list of sample ids and a list of extremely interesting strings such as HOTEIS NOVOS or trgt889. We can deduce that these are the names of

botnets used so once we list sample ids to SHA256 hashes we can identify what APT groups use what botnets and whether or not there are any intersections.

- **C2:** The C2 dataset contains a list of sample ids and IP addresses. We can presume that the samples have been associated with these IP addresses as they use them as C2 servers. Hopefully, we can identify whether two or more APT groups use the same C2 server suggesting collaboration.
- **Credentials:** The credentials dataset contains several pieces of information all regarding what we can assume is account information that these samples take advantage of. Some information included are hosts, protocols, ports, and usernames.
- **Family:** The family dataset contains information that links sample ids to malware families such as emotet or agenttesla.
- **ATT&CK:** The ATT&CK dataset is considered our most important threat intelligence. the MITRE data is extremely useful in portraying how an APT group operates by linking its activities to ATT&CK techniques and tactics.
- **IoCs:** We are unsure whether or not the IoCs data will provide any useful or unique information as they contain a list of IP addresses. Both this and the C2 data were taken from different objects in the JSON array however we are unsure whether or not they are identical. The use of IoCs tells us that these IP addresses could be considered as an indicator that this specific malware has infected a system whereas C2 server IP addresses from what we can tell are servers used by the threat actor.

3.2 Data Sets

The collection of data from VX-Underground, VirusTotal and Triage has allowed us to create a wide range of datasets fine-tuned to the specific areas of our research. For example, if we wanted to provide a timeline of APT groups or attempt to identify any patterns concerning the platform type then we will be able to rely on the data collected from VirusTotal or if we wanted to analyse the families associated with the SHA256 hashes then we can use data collected from VX-Underground and Triage. We can see from the tables below that the SHA256 hash is omnipresent and will be used as the key that links all the datasets together. Figure 7 shows a mockup of all our datasets and how they interact with each other. Blue represents datasets from VirusTotal, orange from Triage and green from VX-Underground. As previously mentioned the datasets are all joined by SHA256 and have a *one and only one* data properly shown with the arrows lines connecting them. Most datasets share the same properties as those that are connected to them. For example, the

starting dataset *modified_dataset.csv* is connected to all the Triage datasets through the SHA256 hash, this also means that the Triage datasets also share *apt_group* which is also shared with some datasets from VX-Underground that are connected to Triage datasets.

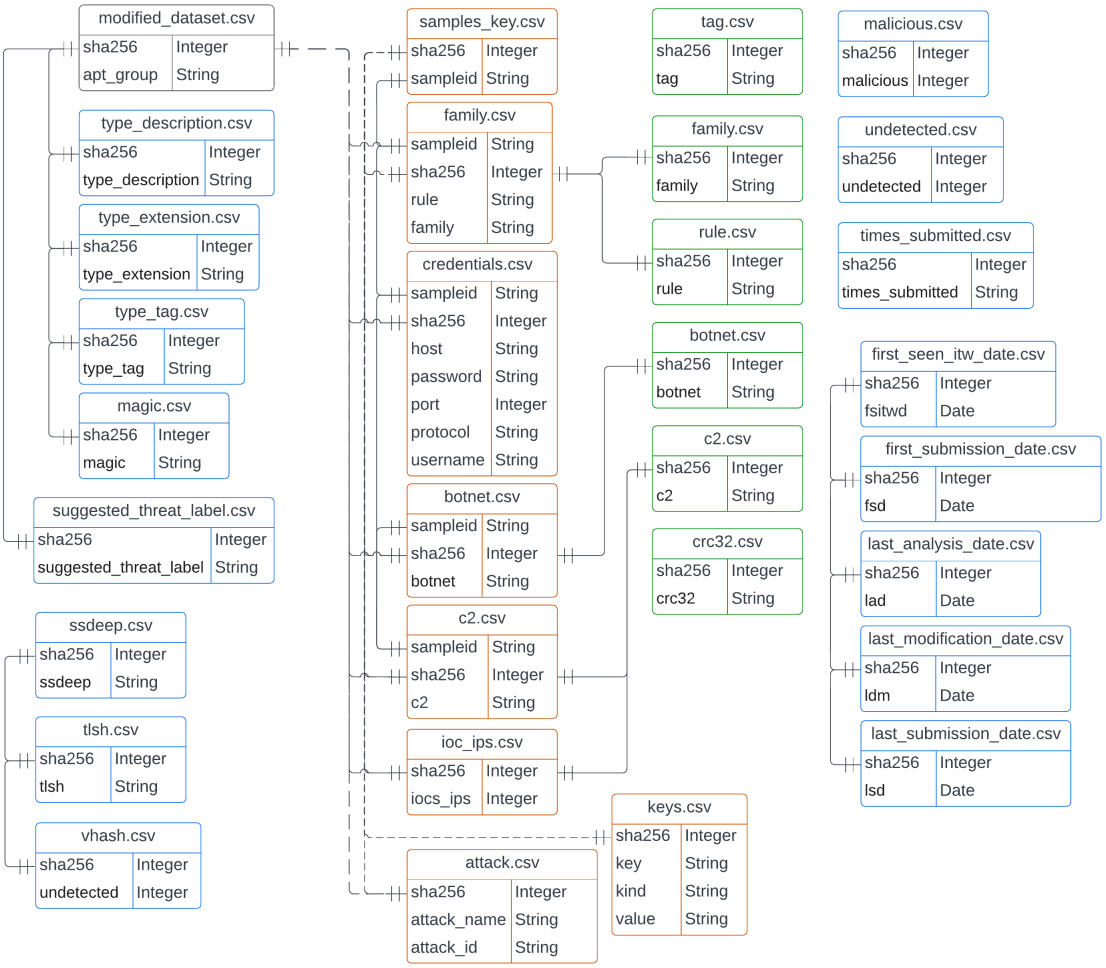


Figure 7: Datasets & schemas

3.2.1 Evaluation against the NIST Criteria

To evaluate the effectiveness of our datasets we can refer to NIST's criteria. All our data was collected from information based on the APT group's campaigns and attacks that have already happened, a consequence of this means that our data doesn't exhibit the timely characteristic. With that being said because our data was collected based on malware that was used means that our intelligence displays the relevant characteristic. Organisations are likely to face these attacks because they have already happened. Our intelligence is extremely accurate, it was collected from reliable security vendors and the data was stored straight away. Based on our quantitative research, our data doesn't exhibit what NIST would state as specific; a story needs to be told for recipients to understand how the threats we have talked about may affect them. Similar to

the previous characteristic, the actionable characteristic would not be met when looking at our data alone. A story involving our data needs to be told for the recipients to understand what countermeasures they can take and what is a suitable response.

3.3 Kaggle

Before any data manipulation and analysis happened we first had to decide what platform we should use. The choices were Jupyter Notebooks or Kaggle; both of them allow us to write in Python and use data science libraries such as seaborn and matplotlib but the biggest factor we considered was usability and collaboratively. Jupyter Notebooks works on a single host computer whereas Kaggle is an online version allowing me to invite those who collaborated on this project.

Firstly we created an online dataset on Kaggle that we uploaded all our CSV files then we created a notebook on Kaggle where we started to create the cells containing the code we used to analyse our data. We used ***tidy_split***, a utility script sourced from the European Union Agency for Cybersecurity. The python script had to be downloaded and then imported into Kaggle for us to be able to use it throughout this project.

4 ANALYSIS AND FINDINGS

4.0.1 Group's Format Speciality

The first pieces of data collected throughout this project were those regarding the malware's file format. Collected from VirusTotal, we were able to identify based on the SHA256 we could identify how APT groups attempt to hide and deliver the malware used in their campaigns. The most common occurrence found in our data was Windows 32 executables. As seen in figure 8, 61.8% of the file types that APT groups used in our data were Win32 EXE, The clear dominance of Win32 EXE is also established in figure 9. 20,478 of our samples were Win32 EXE which is seen in table 4.1.

To provide efficient CTI with this data, we can combine it with other information we have collected. For example, looking at this information and their respective APT groups helps us identify how they operate. XDSpy, an extremely rare APT group that has gone undetected for nine years and has been active since 2011 while stealing secrets from government entities in Eastern Europe including militaries and Ministries of Foreign Affairs (Faou (2020)) has used Win32 EXE and VBA files for their delivery mechanism. To provide a more in-depth analysis we would have to bring in other information such as malware families and that way we would be able to identify what APT group uses what file format that hides a specific piece of malware identified by the family which then allows us to investigate their TTPs.

For additional information, we shouldn't only look at those files most commonly used. 5.89% of our samples were Android applications, we can presume that generating a payload or concealing pre-made malware as an Android Package Kit (APK) would be more complex. Not to mention, that using a malicious APK file would greatly decrease the attack surface unless they were aware that their targets were more susceptible to this method of attack during the reconnaissance stage. For example, StrongPity compromised the official Syrian E-Gov website and replaced the official Android application file with a trojanised version of the original app that would steal contact lists and collect files with specific file extensions from its victim's device (Dong et al. (2021)).

When looking at this information through the eyes of how effective and valuable it is we will need to refer to the literature on CTI. We had this information from files left behind on their victim's devices meaning that knowing this information falls under ***Annoying*** on the PoP.

We can also look toward the criteria outlined by NIST. Considering that we have this data after the APT group's campaign this information can't be considered timely. It can be relevant as organisations can look at their assets and reveal any attack surface that might be used. E.g. whether or not they have an Andriod application. This information can be considered accurate as it was collected as stated by VirusTotal and not altered by me after collection. Knowing whether this information is specific is a difficult choice; it is extremely specific when telling us what file formats were used by what APT groups. However, effective specific CTI according to NIST should allow the recipients of this intelligence a way to understand how the threat may affect them and knowing only what file formats were used doesn't. This intelligence could be considered actionable depending on the situation; knowing that an APT group attacks specific vectors in specific countries using specific file formats could allow an organisation that is a potential target according to their country of origin and their business area to take the steps necessary.

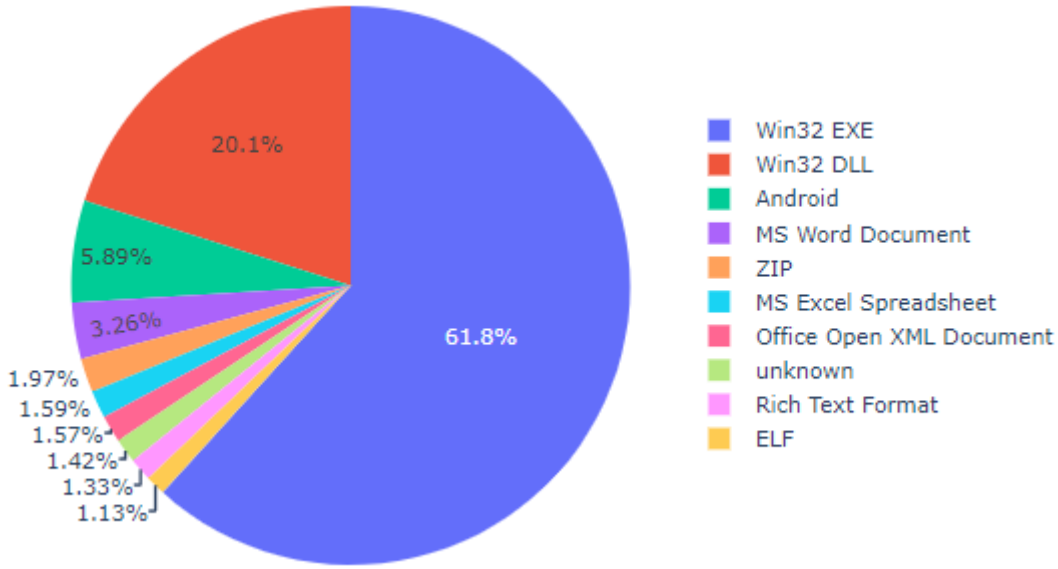


Figure 8: Percentage of File Types.

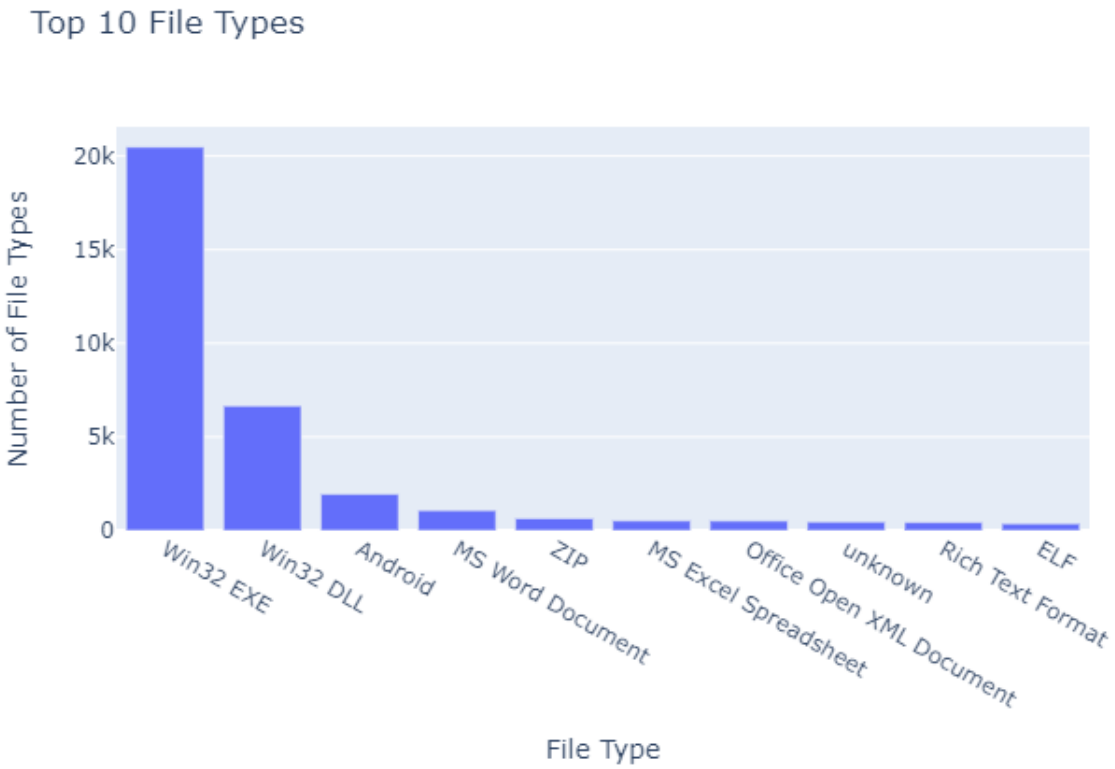


Figure 9: Top 10 Common File Types Occurrence.

Type	Count
Win32 EXE	20,478
Win32 DLL	6,661
Android	1,954
MS Word Document	1,080
ZIP	653
MS Excel Spreadsheet	527
Office Open XML Document	519
unknown	471
Rich Text Format	441
ELF	375

Table 4.1: File Types Count.

4.0.2 Suggested Threat Label

Vendors that are on VirusTotal identify whether or not a file is malicious and what it is and stores this information for other researchers to see. This is why we were able to collect information based on the threat label suggested by these vendors and other researchers when analysing a specific

SHA256 hash. This information, sparsed as it is, can provide useful and effective CTI especially when joined with other information that we have collected such as the APT group associated with the specific sample and the file format used.

The most common occurrence was *trojan.valyria/w97m* seen in figure 10 and table 4.2. This data by itself provides little intelligence but is a brilliant starting point for further research. Valyria is a password-protected Microsoft document file containing malicious scripts embedded in its document body. We have also identified that the APT group Lorec53 uses the samples associated with this threat label suggesting that Lorec53 uses Valyria as malware on their campaigns. Further research into Lorec53 shows us that they're an active Russian hacker group known for phishing attacks against the Georgian Government (Ji (2022)) which aligns with their method of attacking malicious Microsoft documents.

An article by Infoblox confirms that Valyria delivers and drops Emotet onto the victim's computer and reported delivering Agent Tesla, Lokibot, and Kriptik, among others (Ware (2021)). Knowing this information falls under ***Tough!*** on the PoP as we are now aware that Lorec53 uses Valyria to drop Emotet and other malware we can now research Valyria and the other malware to identify TTPs and counter measurements that organisations can take.

We can also look at this attack in terms of the CKCs that we have researched. We can presume that an attack using Valyria would follow the CKCs as such:

- Valyria is delivered through a malicious Microsoft document. (Stage 3 ***Delivery*** on the LKC and the UKC).
- The vulnerability of a human is exploited when they attempt to open the malicious file and executes the malware. (Stage 4 on the LKC and stage 5 on the UKC ***Exploitation***).
- Valyria, once executed, drops Emotet and other malware onto the victim's computer. (Stage 5 on the LKC ***Installation*** and stage 12 on the UKC ***Execution***).
- The malware then changes the victim's computer's registry to connect to their C2 server. (Stage 6 on the LKC and stage 8 on the UKC ***Command & Control***).

We should also refer to the characteristics that CTI would exhibit to be effective. Unfortunately, this data was available to us after the attack. It is extremely relevant as this method of attack is extremely common. The data we have collected is complete and accurate and has not been tampered with after collection. Knowing what malware Valyria drops allows recipients of this information an easy way of understanding how the malware might affect them and we can also research each malware dropped and Valyria itself to look for any countermeasures.

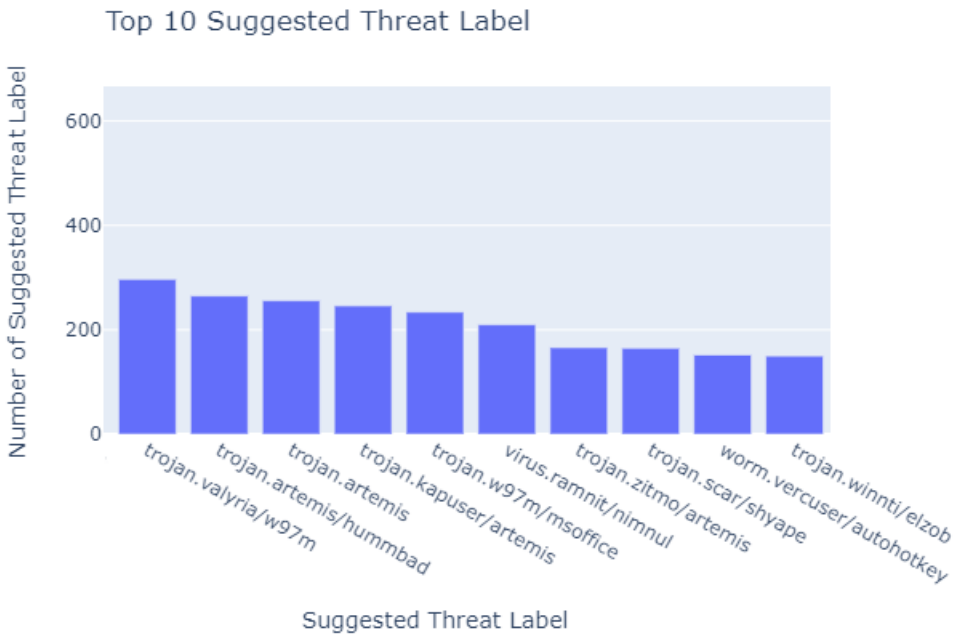


Figure 10: Top 10 Suggested Threat Label Occurrences.

Suggested Threat Label	Count
trojan.valyria/w97m	297
trojan.artemis/hummbad	265
trojan.artemis	256
trojan.kapuser/artemis	246
trojan.w97m/msoffice	234
virus.ramnit/nimnul	210
trojan.zitmo/artemis	166
trojan.scar/shyape	165
worm.vercuser/autohotkey	152
trojan.winnti/elzob	150

Table 4.2: Suggested Threat Label Count.

4.0.3 MITRE ATT&CK

One of the most pivotal pieces of intelligence we wanted to gather and analyse was adversaries' techniques as mapped by MITRE's ATT&CK framework. The reason why it was important to identify these techniques is that it provides us with the TTPs that APT groups use. The most common occurrence was ***T1082: System Information Discovery*** shown in figure 11. Further research into this technique informs us that 23,983 of our samples were techniques assigned to the TTPs when adversaries attempt to get detailed information about the victim's operating system and hardware,

including versions, services packs and architecture (MITRE (2017b)) seen in table 4.3. Additional information from MITRE tells us that this technique can also be associated with malware such as Agent Tesla which seems to provide credential harvesting as the core functionality, geolocation, persistence, keylogger, as well as screen capturing, are optional (Valsamakis (2021)). Knowing TTPs come under ***Tough!*** on the PoP as adversaries will have to go about their attacks differently if we know how to counter them.

MITRE's ATT&CK techniques also make it incredibly easy to map these attacks to stages on the CKCs we mentioned. Research shows that the technique ***T1082: System Information Discovery*** would be classed as both stages of the LKC and UKC ***Reconnaissance***. The next common occurrence was ***T1112: Modify Registry*** which information from MITRE informs us that this is when an APT group interacts with the victim's computer's Windows Registry to hide configuration information within Registry keys, removes information as part of cleaning up, or as part of other techniques to aid in persistence and execution (MITRE (2017a)). We can justify that this technique can fall under more than a single stage of the CKCs. For example, stage 6 ***Persistence***, stage 12 ***Execution*** and stage 16 ***Exfiltration*** of the UKC.

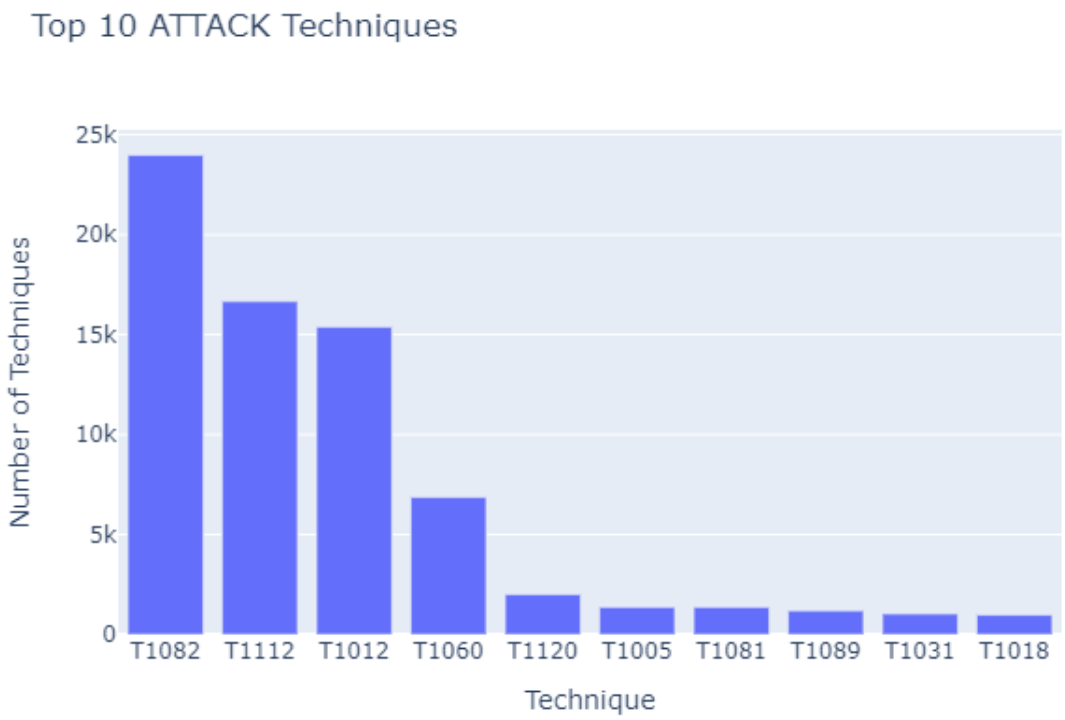


Figure 11: Top 10 ATT&CK Techniques Occurrences.

Technique ID	Name	Count
T1082	System Information Discovery	23,983
T1112	Modify Registry	16,664
T1012	Query Registry	15,388
T1060 (Now T1547)	Registry Run Keys / Startup Folder	6,868
T1120	Peripheral Device Discovery	2004
T1005	Data from Local System	1,371
T1081 (Now T1552)	Credentials in Files	1,371
T1089 (Now T1562)	Disabling Security Tools	1,193
T1031 (Now T1543)	Modify Existing Service	1,035
T1018	Remote System Discovery	979

Table 4.3: ATT&CK Techniques Count.

4.0.4 YARA Rules

YARA rules were another piece of pivotal data that we wanted to collect. These rules contain descriptions of targeted malware based on strings or byte sequences found in the malware. We can utilise these rules to discover malicious files or processes (Naik et al. (2019)). More importantly, we can use these rules to help identify the malware used and how they work. You can see from figure 12 that the most common YARA rule that was found in our data was ispe32 which appears to be the YARA rule assigned to malware that is portable executables running on 32-bit Window operating systems which corresponds with previous findings from the file formats. One thing that should be noted about this data is that it includes some YARA rules that might not be of any interest and are capabilities that we were already aware of.

Figure 12: Top 10 YARA Rules Occurrences.

To further our analysis of YARA rules, we achieved this by returning the top 40 rules based on the number of occurrences within our data, we then decided to select the rules that provided the more interesting analysis; rules such as ***yara:seh_save*** and ***yara:anti_dbg***. As you can see from figure 13, these rules were picked out of the list of 40 rules as they are the most interesting and informative when looking at them as CTI. 20,191 of our samples were associated to the YARA rule ***yara:anti_dbg***, 8,293 of them to ***yara:keylogger*** and 6,868 of them to ***yara:escalate_priv***.

Looking at how effective this CTI is, we can refer back to what characteristics are needed; these samples are already in the wild and are being studied so they don't exhibit timely. They're incredibly relevant as they are referring to samples that have embedded key loggers and can escalate the adversary's privileges; both are incredibly important and likely to happen during a cyber attack. Thanks to the nature of YARA rules the CTI is accurate and extremely specific.

With the above YARA rules, we can also refer to other CTI criteria. For example, on the UKC, the YARA rule ***yara:escalate_priv*** could reference stage 11 ***Privilege Escalation*** where the adversary will attempt to access higher permissions in their victim's system. The rule ***yara:keylogger*** could reference stage 13 ***Credential Access*** where the attacker retrieves system credentials if the victim were unaware and typed in their credentials. Referencing the PoP, knowing these rules fall on ***Tough!*** of the pyramid. knowing these capabilities will cause the attackers the most pain since we are aware of how the malware works and can take steps to prevent it causing the attackers to have to go about their attacks differently.

Figure 13: Interesting YARA Rules.

Knowing how many occurrences we have of each rule is interesting in itself but when combined with APT groups we can begin to see some interesting results. We fed this data into CosineSimilarity.py seen in D.0.5 that creates a vector for each APT group and counts how many times each rule was used; values are from 0 to 1. The closer the value is to 1, the more closely linked the APT groups are. For example, table 4.4 shows some of our results including the APTs closest to 1 and some being furthermost away. With this data, we should be able to research and validate others' intelligence and find possible correlations and collaboration among campaigns and threat actors.

A threat alert was released by Rewterz (2022) regarding the APT groups: Phosphorus and TunnelVision listing some IoCs and the Impact that was caused by this attack and stating that "Some vendors suggest that the attributes of TunnelVision and Phosphorus are similar and therefore they must be linked, however, current data is insufficient to confirm this theory". Our data corroborate that these two APT groups are linked. Another article released by Cyware (2022b) states that "The TTPs of TunnelVision overlap with Iran-linked nation-state actor APT35, also known as Phosphorus, Nemesis Kitten, and Charming Kitten." Our data has the APT group TunnelVision being linked to Charming Kitten which corroborates what Cyware has stated. In addition, our data also has Charming Kitten linked to Phosphorus suggesting that these three APT groups are collaborating. A report released by Cyware (2021) regarding the APT group Evil Corp where tell analyse the group as a whole including their motive, targeted countries and sectors alongside their TTPs. One of Evil Corp's alias is TA505, another success that our data corroborates.

It's also important to showcase if any of our results were not found to be linked. We researched the supposedly linked APT groups: APT 1 and Penquin Turla. Mandiant presents a large amount of evidence that the APT 1 threat group comes from China, namely the People's Liberation Army Unit 61398 (Wangen (2015)), whereas Turla, is the work of the Russian government. There is no clear link between these two groups but that is not to say that there is none, we should also take into consideration that the similarity was 0.88. Another low similarity included in our data was for

the Darkhotel; allegedly linked to South Korea by the Chinese cyber security company Qihoo 360 (Delerue (2021)) and Pitty Panda; A known Chinese threat group (Kim et al. (2021)) with a score of 0.89. Similar to the other low similarity pairs of APT groups, there is no clear literature that supports a link.

From the examples above, it should be clear that this data and methodology can be extremely useful to guide an investigation. An endpoint user such as a SOC analyst can understand the YARA rules and their associations to an APT group and then use this knowledge to determine how closely a link an APT group is to their investigation. We can consider a scenario where an organisation notices orphan YARA rules with no APT attribution yet where the organisation can create a placeholder and run the script. Another scenario could be that a SOC analyst for the company finds proof of a known APT and can be directly linked with the other APT groups from table 4.4. The SOC analyst can now receive a more effective CTI and act on it.

First APT	Second APT	Similarity	Literature Connection
APT 19	Shell Crew	1.0	Y
APT-C-09	Dropping Elephant	1.0	Y
APT-C-09	Patchwork	1.0	Y
Dropping Elephant	Patchwork	1.0	Y
Charming Kitten	Phosphorous	1.0	Y
Charming Kitten	TunnelVision	1.0	Y
Phosphorous	TunnelVision	1.0	Y
Evilcorp	TA505	1.0	Y
MuddyWater	Nobelium	0.9	N
Mustang Panda	StrongPity	0.9	N
APT 29	MuddyWater	0.9	N
APT 1	Penquin Turla	0.88	N
Darkhotel	Pitty Tiger	0.89	N
APT 10	Syrian Electronic Army	0.89	N

Table 4.4: Cosine Similarity of YARA Rules & APT Groups.

Figure 14 shows a default network diagram that was created using the above data. The diagram shows interconnections between our APT groups. each APT group is represented by a node and the connections between are the similarity score from the cosine difference metric.

The clarity of the diagram portrays an easy-to-follow story of attribution linkage between our APT groups. The diagram suggests that APT 23 has been linked to antlion. Interestingly an employee

of Curated Intelligence going by the Twitter handle ”Arkbird” released his notes onto GitHub stating ”Despite the few samples available, some notes on relations between #APT23 and #Antlion could be found but difficult to confirm at 100%.” (Arkbird (2022)). It is unknown whether the data that we have confirms for definite that there is a link but a similarity score of 0.99 clearly shows that both these groups have used the same TTPs and the YARA rules associated with them.

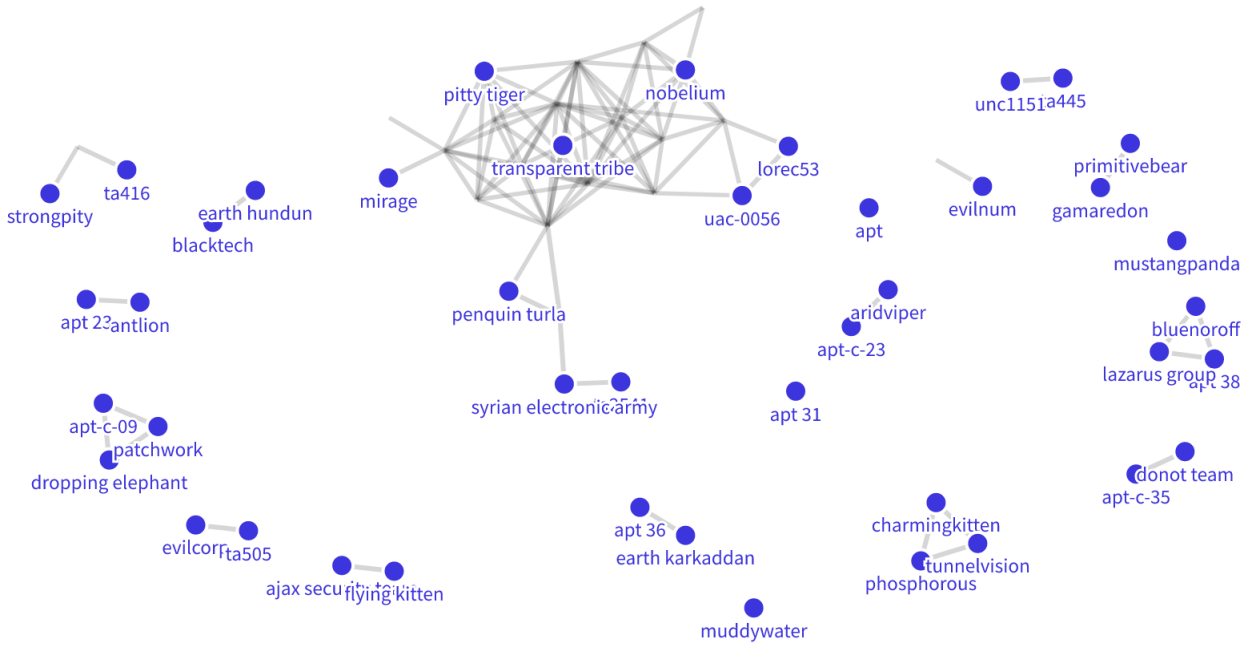


Figure 14: APT Attribution and Linkage.

4.0.5 Malware Families

Malware families were another piece of data that we were able to retrieve from both VirusTotal and Triage. These families were associated with the SHA256 hashes and informed us what malware families that they belong to. Combining this data with APT groups can provide us with what malware an APT group uses and from there we can look into how that malware operates and its TTPs. The most common malware family that was in our data and seen in figure 15 was trickbot. Trickbot is malware aimed at stealing valuable information from the targets that it infects, trickbot is well known for its ability to steal passwords saved in browsers (Gittins and Soltys (2020)). The main limitation of this data was completeness. The aforementioned trickbot malware being the most common only had 258 hits. The extremely low number calls into question the usability of this data set and makes it unlikely to provide actionable threat intelligence based on an APT group’s MO.

As we previously mentioned knowing what malware an APT group use could inform us what tools

and TTPs are associated with the threat actor. This information could be said to fall on the upper two layers of the pyramid: ***Tough!*** and ***Challenging***.

We know that Trickbot can (Farooghian et al. (2020)):

- Steal sensitive information, including banking login details and memorable information, steal login credentials for infected devices, including domain credentials and steal saved online account passwords, cookies and web history. (Stage 13 ***Credential Access*** on the UKC).
- Gather detailed information about infected devices and networks. (Stage 15 ***Collection*** on the UKC).
- Connect infected devices to malicious, criminally-controlled networks over the internet, giving criminals full control of them. (Stage 6 ***Command & Control (C2)*** on the LKC and stage 14 ***Lateral Movement***).
- Download further malicious files such as Remote Access Tools, VNC clients and ransomware. (Stage 5 ***Installation*** on the LKC).

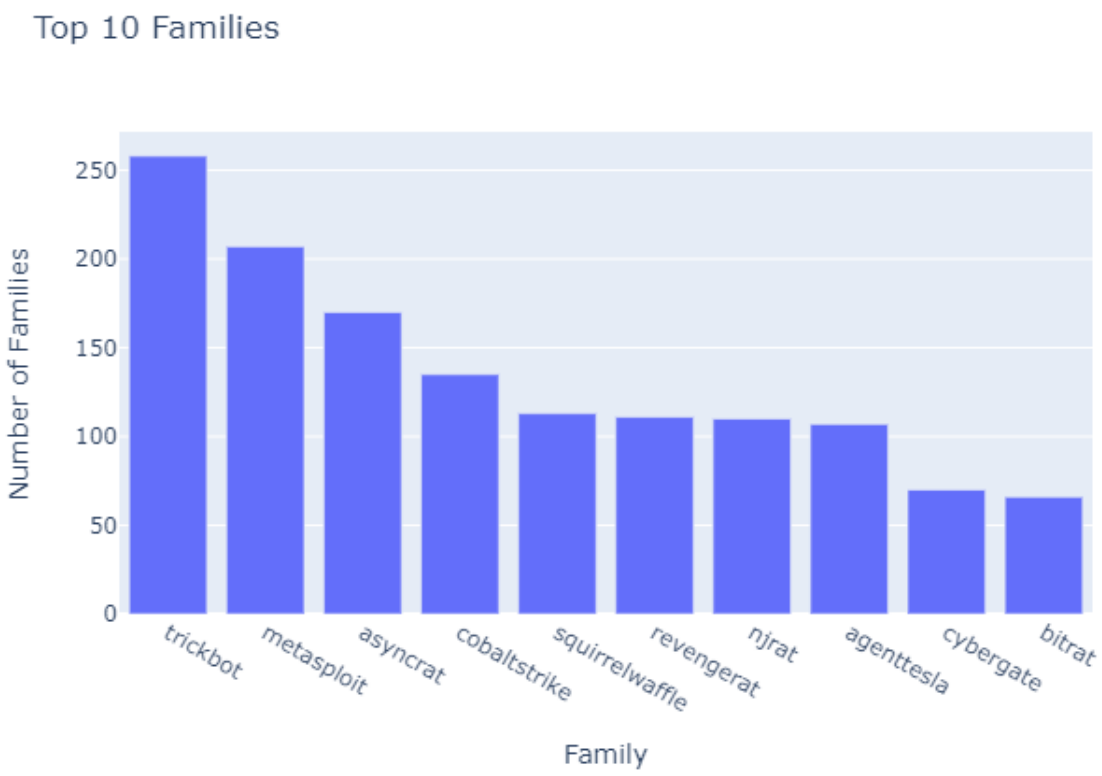


Figure 15: Top 10 Malware Families Occurrences.

5 CONCLUSION

5.1 Research Questions

5.1.1 Research Question 1

In RQ_1 we asked whether it was feasible to use low-value IoCs such as SHA256 hashes to construct an attack's CKC. We found that these hashes alone were not useful, that having been said, the CTI that we can get from these hashes can certainly provide us with the information needed to construct an attack's CKC.

A vital piece of data was ATT&CK techniques associated with these hashes. One of the top common occurrences in our data was ***T1082: System Information Discovery***. Further researching this technique using the data stored on MITRE, we discovered that this technique is for when an APT group attempts to get detailed information about the operating system of their victims. The very nature of this technique could be considered associated with the reconnaissance stage of both CKCs we will discuss. We also found that ***T1112: Modify Registry*** was another common occurrence and further information shows that we can construct a CKC including stage 6 ***Persistence***, stage 12 ***Execution*** and stage 16 ***Exfiltration*** of the UKC.

5.1.2 Research Question 2

RQ_2 asks whether malware signatures are sufficient IoCs to perform APT attribution. We found that YARA rules signatures used to identify malware capabilities were the best way to perform APT attribution. During our analysis of our YARA rules, we performed a cosine similarity test with these rules to show the confidence that two APT groups were linked based on whether they used the same TTPs identified by the YARA rules. We can be confident in APT attribution and collaboration of two groups based on the similarity metric.

5.1.3 Research Question 3

In RQ_3 we asked whether it was possible to provide efficient CTI using the information that we have that is sparsed. Not all the data that we collected was rich in quantity, the data we collected regarding the malware families that have been associated with a SHA256 was interesting but was extremely sparsed with the most common family was ***trickbot*** with just over 250 occurrences.

That having been said, when combined with our other data sets it provided an extremely in-depth look into APT groups case by case.

AsyncRAT, a tool developed in C# language, was originally designed to monitor computers remotely via secured connections. The tool has been exploited maliciously as a RAT by the threat actors globally owing to its vast capabilities, such as keylogging, audio/video recording, info-stealing, remote desktop controlling, and more. (Cyware (2022a)). This was one of the families found in our data and we can determine that the APT group TA2541 uses this RAT during their campaigns and attacks. Upon further research, we can see TA2541's motives, whom they target and what sectors they hit. We can also determine their TTPs based on information we found on AsyncRAT. We can see that TA2541 recent campaigns have them delivering their RAT through obfuscated Visual Basic Script (VBS) files. AsyncRAT then adds the VBS file into the victim's startup directory establishing persistence. Interesting the threat report released by Larson and Wise (2022) from proofpoint links information from our datasets together by establishing TA2541's method of delivery and mentioning the term "kimjoy" which is found in sever of their C2 domain names.

5.1.4 Research Question 4

RQ_4 is an interesting question which would require deduction reasoning. We asked whether it was possible to predict future targets based on the CTI that we have collected. When all our data sets have been merged it provides an in-depth history of an APT group and their attacks.

The CTI that we would look at would be the APT group where we would investigate their motives and preferred targets for both victim and victim's sector. We would also look at the APT's MO by the families that they use and the file formats that they have used in their previous attacks. We then have to use deductive reasoning to take all this data and turn it into a story.

5.2 Future Work

The original list of samples that we collected was over 600,000. This was reduced to only include the samples that we could identify the APT group associated using VX-Underground. Any future would that would carry on what we set out to investigate here would include all samples. We then could spend more time analysing each sample by using other tools to have a bigger data pool. This would have been brilliant for our YARA rules analysis.

I was made aware of a CAPA developed by Mandiant that can detect capabilities in executable files. The obvious problem with this is that CAPA works by analysing the executable which would not be advisable as we would have to download and store the files, we would need to consider

how we would safely store all these files.

We have also discussed implementing Natural Language Processing where we program and process threat reports released by reputable vendors like FireEye and Mandiant where they would be read and retrieve any useful information like IoCs which will then be compared with our dataset. This will allow users to perform a deeper search and discover any correlations between APT groups.

5.3 Limitations & Caveats

One of the main factors that had to be sorted regarding our usage of APIs was the keys. A public key from VirusTotal allowed us to submit 4 requests a minute, 500 requests a day and 15,500 requests a month. An email exchange with VirusTotal's business team allowed me to explain why and how I would use an academic key; shortly after my account was upgraded so we never faced the issue of the 15,500 requests. That being said, this new key allowed us to make roughly 50,000 requests a day which we achieved multiple times. This project would not have been viable without this key as what took a week would've taken months.

Each script ran returned the data we wanted in JSON format. This was potentially another problem as CSVs were required to use Kaggle and it made manually sorting and fixing some data easier. The main issue with this came when I attempted to convert from JSON to CSV; Microsoft Excel had a simple feature that would have done exactly what I needed if it wasn't for the fact that a small majority of the 28,000 JSON files had been duplicated during the collection that causes Excel to throw the error "DataFormat.Error". Upon asking the Unix and Linux community for advice on how to automate the cleaning I was provided with Jq, a Linux CLI tool that automated removing the duplicate JSON object. The script "jq.sh" used can be seen in D.0.4.

Our analysis of YARA rules using cosine similarities provided an interesting limitation. With our lower similarity value, we found that some of the APT pairs were not found to be linked by any available literature which meant that they were a false positive. These incorrect APT pairs could infect our analysis of the correct results. Further investigation into the pairs with a value less than 1 would be needed to ensure that we are correct in saying that they're not linked. Even then, these incorrect pairs still add value to CTI.

Word count: 10,000

Bibliography

Al-Mohannadi, H., Mirza, Q., Namanya, A., Awan, I., Cullen, A. and Disso, J., 2016. Cyber-attack modeling analysis techniques: An overview. *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 69–76.

Alshamrani, A., Myneni, S., Chowdhary, A. and Huang, D., 2019. A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys & Tutorials*, 21 (2), 1851–1877.

Apps, P. and Finkle, J., 2014. Suspected russian spyware turla targets europe, united states. *Reuters. com US Edition, March*, 7. URL `https://paper.seebug.org/papers/APT/APT_CyberCriminal_Campagin/2014/Reuters_Turla.pdf`.

Arkbird, 2022. URL `https://twitter.com/JAMESWT_MHT/status/1491427528053227520`.

Baker, K., 2022. What is cyber threat intelligence? URL `https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/`.

Chierzi, V. and Merces, F., 2021. Evolution of iot linux malware: A mitre att&ck ttp based approach. *2021 APWG Symposium on Electronic Crime Research (eCrime)*, 1–11.

Cyware, 2021. Evil corp - a threat actor with multiple alter egos: Cyware: Research and analysis. URL `https://cyware.com/research-and-analysis/evil-corp-a-threat-actor-with-multiple-alter-egos-de7c`.

Cyware, 2022a. Asyncrat: The anatomy of a highly-evasive malware: Cyware: Research and analysis. URL `https://cyware.com/research-and-analysis/asyncrat-the-anatomy-of-a-highly-evasive-malware-fa51`.

Cyware, 2022b. Tunnelvision apt group exploits log4shell: Cyware hacker news. URL `https://cyware.com/news/tunnelvision-apt-group-exploits-log4shell-aa638405`.

Das Purba, M., Chu, B. and Al-Shaer, E., 2020. From word embedding to cyber-phrase embedding: Comparison of processing cybersecurity texts. *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 1–6.

Delerue, F., 2021. Covid-19 and the cyber pandemic: A plea for international law and the rule of sovereignty in cyberspace. *2021 13th International Conference on Cyber Conflict (CyCon)*, 9–24.

Dong, Z., Yarochkin, F. and Du, S., 2021. Strongpity apt group deploys android malware for the first time. URL https://www.trendmicro.com/en_us/research/21/g/strongpity-apt-group-deploys-android-malware-for-the-first-time.html.

Faou, M., 2020. URL https://www.welivesecurity.com/2020/10/02/xdspy-stealing-government-secrets-since-2011/.

Farooghian, S., Kivimagi, K. and Lovell, K. N., 2020. *HANDS-ON-SKILLS LAB ON OPEN-SOURCE INTRUSION DETECTION SYSTEM(IDS)*. Ph.D. thesis, Tallinn University of Technology.

Gittins, Z. and Soltys, M., 2020. Malware persistence mechanisms. *Procedia Computer Science*, 176, 88–97. URL https://www.sciencedirect.com/science/article/pii/S1877050920318342, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.

Hatching, 2018. Welcome to triage. URL https://tria.ge/.

Herring, M., n.d. Threat detection and the pyramid of pain. URL https://www.criticalstart.com/threat-detection-and-the-pyramid-of-pain/.

Hutchins, E. M., Cloppert, M. J., Amin, R. M. et al., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1 (1), 80.

Jasper, S. E., 2019. North korea's cyberspace aggression. *International Journal of Intelligence and CounterIntelligence*, 32 (1), 194–198. URL https://doi.org/10.1080/08850607.2018.1524247.

Ji, J., 2022. Apt retrospection: Lorec53, an active russian hack group launched phishing attacks against georgian government. URL https://nsfocusglobal.com/apt-retrospection-lorec53-an-active-russian-hack-group-launched-phishing-attacks-against-

Johnson, C., Badger, L., Waltermire, D., Snyder, J., Skorupka, C. et al., 2016. Guide to cyber threat information sharing. *NIST special publication*, 800 (150). URL https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-150.pdf.

Kim, K., Alfouzan, F. A. and Kim, H., 2021. Cyber-attack scoring model based on the offensive cybersecurity framework. *Applied Sciences*, 11 (16). URL https://www.mdpi.com/2076-3417/11/16/7738.

Kim, Y.-K., Lee, J. J., Go, M.-H. and Lee, K., 2020. Analysis of the asymmetrical relationships between state actors and apt threat groups. *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 695–700.

Larson, S. and Wise, J., 2022. Charting ta2541's flight. URL https://www.proofpoint.com/uk/blog/threat-insight/charting-ta2541s-flight.

Liu, X., Li, L., Ma, Z., Lin, X. and Cao, J., 2019. Design of apt attack defense system based on dynamic deception. *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, 1655–1659.

MITRE, 1997. Situation awareness. URL https://www.mitre.org/capabilities/cybersecurity/situation-awareness.

MITRE, 2017a. Modify registry. URL https://attack.mitre.org/techniques/T1112/.

MITRE, 2017b. System information discovery. URL https://attack.mitre.org/techniques/T1082/.

Naik, N., Jenkins, P., Savage, N. and Yang, L., 2019. Cyberthreat hunting - part 1: Triaging ransomware using fuzzy hashing, import hashing and yara rules. *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–6.

Narayanan, M., 2021. Forestalling a cyber pearl harbour. *The Hindu*. URL https://www.hindujagroup.com/admin/upload/HGSpecialFeatures/Attachments/SpecialFeatures_637521156926828285.pdf.

Okada, S., Fujiwara, Y., Fujimoto, M., Matsuda, W. and Mitsunaga, T., 2021. Efficient incident response system on shared cyber threat information using sdn and stix. *2021 IEEE International Conference on Computing (ICOCO)*, 109–114.

Pols, P. and van den Berg, J., 2017. The unified kill chain. *CSA Thesis, Hague*, 1–104.

Rewterz, 2022. Rewterz threat alert – phosphorus/tunnelvision apt – active iocs. URL https://www.rewterz.com/rewterz-news/rewterz-threat-alert-phosphorus-tunnelvision-apt-active-iocs.

Rogers, D. and Kanth, V., 2021. Wannacry data collection system design and tutorial. *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 1–6.

Satheesh Kumar, M., Ben-Othman, J. and Srinivasagan, K., 2018. An investigation on wannacry ransomware and its detection. *2018 IEEE Symposium on Computers and Communications (ISCC)*, 1–6.

Smilyanets, D., 2022. How vx-underground is building a hacker's dream library. URL https://therecord.media/how-vx-underground-is-building-a-hackers-dream-library/.

STIX, 2017. Introduction to stix. URL https://oasis-open.github.io/cti-documentation/stix/intro.

Valsamakis, K., 2021. *Windows Malware Analysis*. Ph.D. thesis, University of Piraeus.

Virustotal, 2004. Virustotal. URL https://www.virustotal.com/.

Wangen, G., 2015. The role of malware in reported cyber espionage: A review of the impact and mechanism. *Information*, 6 (2), 183–211. URL https://www.mdpi.com/2078-2489/6/2/183.

Ware, J., 2021. Valyria trojan drops emotet. URL https://blogs.infoblox.com/cyber-threat-intelligence/cyber-campaign-briefs/valyria-trojan-drops-emotet/.

Warikoo, A., 2021. The triangle model for cyber threat attribution. *Journal of Cyber Security Technology*, 5 (3-4), 191–208. URL https://doi.org/10.1080/23742917.2021.1895532.

Zhang, P., Ya, J., Liu, T., Li, Q., Shi, J. and Gu, Z., 2019. imcircle: Automatic mining of indicators of compromise from the web. *2019 IEEE Symposium on Computers and Communications (ISCC)*, 1–6.

Zhang, Q., Li, H. and Hu, J., 2017. A study on security framework against advanced persistent threat. *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 128–131.

Appendix A - Project Proposal

Bournemouth University

Department of Computing and Informatics

2021-22AY Level 7 Post-Graduate Individual Project

Project Proposal Form

Please refer to the **Project Handbook** when completing this form and note that your proposal should be your own original work and you must cite sources in line with university guidance on **referencing and plagiarism**[1].

Student's Name:	**Paul-David Jarvis**
Degree Title:	**MSc Internet of Things with Cyber Security**
Project Title/Area:	**APT Groups**
Supervisor's Name:	**Vasilis Katos**

Section 1: Project Overview

1.1 Problem definition - use one sentence to summarise the problem:

Analysing APT groups and malware they used.

1.2 Project description - briefly explain your project:

Creating And analysing a large data set containing information on APT groups, malware they used and attributions of both for classification purposes.

1.3 Background - please provide brief background information, e.g., client, problem domain, referencing literature (minimum 4-5 sources):

Using VX-Underground (https://www.vx-underground.org/) to collect a large range of malware samples and attributes. Using VirusTotal's (virustotal.com/) API to analyse and scrape attributions of samples. Using data gathered from Triage (https://tria.ge/) and Using Kaggle (https://www.kaggle.com) Notebook to analyse and manipulate our data. Using additional information from several therat intelligence sources like MITRE (https://attack.mitre.org/)

[1] https://libguides.bournemouth.ac.uk/study-skills-referencing-plagiarism

Level 7 Project Proposal Template 2021/2022AY 1

Figure 16: First Page of Project Proposal.

Bournemouth University

Department of Computing and Informatics

2021-22AY Level 7 Post-Graduate Individual Project

1.4 Research question(s):

- What is the format that each group specialises? DLL, EXE, ELF?
- What is the threat category that each group specialises?
- Use first_submission_date from VT to determine patterns of campaigns per group, they are not expected to be working all the time.
- VXUG has the yara rules for the detection of each sample which imply capabilities, methods, strings etc. So the questions are
  - What are the most common rules per group?
  - Which are the most common rules per group intersections?
- VT & Triage provides domain names/IPs. Are there any intersections that imply the use of common infrastructure?
- Can we have the MITRE techniques for each sample from VXUG so that we can verify
- Clustering: using SSDEEP, TLSH, and imphash (where available). Do we have tangible intersections between APT groups

1.5 Aims and objectives – should be specific and measurable:

To collect a large sample of hashes
To collect a large range of attributes, enough to run classifications
Produce reports and datasets based on the above research questions

Section 2: Artefact and Planning

2.1 What is the artefact that you intend to produce?

Dataset and jupyter notebook files, alongside other additional files depending on where the project goes.

2.2 How do you intend on conducting the project (to produce the above artefact successfully)?
Please provide a GANTT chart at the end of this form, in Section 6.

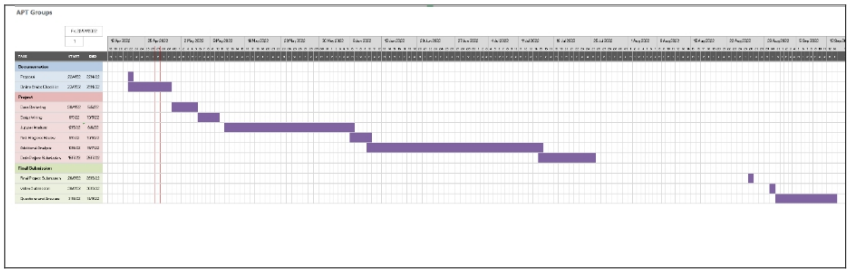
Level 7 Project Proposal Template 2021/2022AY 2

Figure 17: Second Page of Project Proposal.

Bournemouth University

Department of Computing and Informatics

2021-22AY Level 7 Post-Graduate Individual Project



Section 3: Evaluation

3.1 How are you going to evaluate your project artefact?

By determining whether we collected enough samples and enough attributes to be able to answer the above research questions.

3.2 How does this project relate to your MSc Programme?

The title of my degree is Internet of Things with Cyber Security, this project will be about Cyber Security, APT groups, and malicious files.

3.3 What are the risks in this project and how are you going to manage them?

There are zero risks to this project. All data is publicly available.

Section 4: References

Level 7 Project Proposal Template 2021/2022AY 3

Figure 18: Third Page of Project Proposal.

Bournemouth University

Department of Computing and Informatics

2021-22AY Level 7 Post-Graduate Individual Project

4.1 Please provide references for citations used above.

https://www.vx-underground.org/

virustotal.com/

https://tria.ge/

https://jupyter.org/

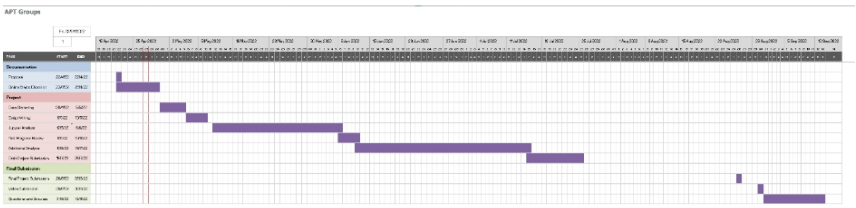
https://attack.mitre.org/

Section 5: Academic Practice and Ethics

Please delete as appropriate.

5.1 Have you made yourself familiar with, and understand, the University guidance on referencing and plagiarism?	**Yes**
5.2 Do you acknowledge that this project proposal is your own work and that it does not contravene any academic offence as specified in the University's regulations?	**Yes**
5.3 Have you submitted online ethics checklist to your supervisor?	**No**
5.4 Has the checklist been approved by your supervisor?	**No**

Section 6: Proposed Plan (please attach a GANTT chart below)



Level 7 Project Proposal Template 2021/2022AY 4

Figure 19: Fourth Page of Project Proposal.

Appendix B - Ethics Checklist

Bournemouth University

Research Ethics Checklist

About Your Checklist	
Ethics ID	41510
Date Created	14/01/2022 14:00:49
Status	Approved
Date Approved	19/05/2022 15:24:05
Date Submitted	18/05/2022 14:46:34
Risk	High

Researcher Details	
Name	Paul-David Jarvis
Faculty	Faculty of Science & Technology
Status	Postgraduate Taught (Masters, MA, MSc, MBA, LLM)
Course	MSc Internet of Things with Cyber Security
Have you received funding to support this research project?	No

Project Details	
Title	APT Analysis
Start Date of Project	16/05/2021
End Date of Project	26/08/2022
Proposed Start Date of Data Collection	16/05/2021
Original Supervisor	Vasilis Katos
Approver	Ethics Programme Team

Summary - no more than 600 words (including detail on background methodology, sample, outcomes, etc.)

This project consists of a collection and analysis of roughly 30,000 unique hashes that are associated to malicious samples from VirusTotal, Triage and VX-Underground; these hashes are not exploits themselves, meaning there is no security risk. The scripts we write scrape the relevant data from the vendor's databases using their application programming interface (API)s to automate this process. These samples have associated to them attribution data such as times they were first seen and Command and control (C2) servers that they communicate with. All this data will then be categorised and used alongside Jupyter labs to manipulate the data to achieve our objectives.

Our objectives are to provide a detail analysis on Advanced Persistent Threat (APT) groups and the malware they use. We hope to further the knowledge known in the community about APT groups and how they operate. The end result and how we are going to portray this data is unclear but the idea of a website that allows users to view this information has been floated.

Page 1 of 2 Printed On 15/08/2022 13:38:58

Figure 20: First Page of Ethics Checklist.

Filter Question: Is your study solely literature based?

Additional Details	
Will you have access to personal data that allows you to identify individuals which is not already in the public domain?	No
Will you have access to confidential corporate or company data (that is not covered by confidentiality terms within an agreement or separate confidentiality agreement)?	No

Storage, Access and Disposal of Research Data	
Once your project completes, will any anonymised research data be stored on BU's Online Research Data Repository "BORDaR"?	No
Please explain why you do not intend to deposit your research data on BORDaR? E.g. do you intend to deposit your research data in another data repository (discipline or funder specific)? If so, please provide details.	
NOT APPLICABLE	

Page 2 of 2 Printed On 15/08/2022 13:38:58

Figure 21: Second Page of Ethics Checklist.

Appendix C - Artifacts

C.0.1 Kaggle Notebook

- **1.0 Group's Format Speciality**
- **2.0 Threat Category**
- **3.0 Timeline**
- **4.0 YARA Rules**
- **5.0 IPs & Domains**
- **6.0 ATT&CK**
- **7.0 Malware Families**
- **Clustering - Future Work**

C.0.2 APT Analysis Datasets

- **VT_first_seen_itw_date.csv**
- **VT_first_submission_date.csv**
- **VT_last_analysis_date.csv**
- **VT_last_modification_date.csv**
- **VT_last_submission_date.csv**
- **VT_magic.csv**
- **VT_malicious.csv**
- **VT_ssdeep.csv**
- **VT_suggested_threat_label.csv**
- **VT_times_submitted.csv**
- **VT_tlsh.csv**
- **VT_type_description.csv**

- **VT_type_extension.csv**
- **VT_type_tag.csv**
- **VT_undetected.csv**
- **VT_vhash.csv**
- **VT_vhash.csv.csv**
- **apt_tags.csv**
- **botnets.csv**
- **c2.csv**
- **dataset.csv**
- **imphash.csv**
- **malware_families.csv**
- **modified_dataset.csv**
- **triage_attack.csv**
- **triage_credentials.csv**
- **triage_iocs_ips.csv**
- **triage_keys.csv**
- **triage_samples_key.csv**
- **vx_crc32.csv**
- **vx_rule.csv**
- **vx_tag.csv**

Appendix D - Scripts

D.0.1 VirusTotal

FileReport.sh

```
#!/bin/bash
while read hashes; do
    echo "Hash:" $hashes
        curl https://www.virustotal.com/api/v3/files/$hashes >>
        /home/jarvis4444/Desktop/VirusTotal/Hashes/$hashes \

        --header 'x-apikey: APIKEY'
 done < hashes
```

D.0.2 VX-Underground

CLI-FileDetails.sh

```
#!/bin/bash
while read hashes; do
    echo $hashes
        mwdb get details -o json $hashes >>
        /home/jarvis4444/Desktop/VX-Underground/CLI/$hashes
done < hashes
```

HTTP-FileDetails.sh

```
#!/bin/bash
while read hashes; do
    echo $hashes
        curl https://virus.exchange/api/file/$hashes >>
        /home/jarvis4444/Desktop/VX-Underground/CLI/$hashes \

        --header 'Authorization: Bearer APIKEY'
```

```
done < hashes
```

D.0.3 Triage

Search.sh

```
#!/bin/bash
while read hashes; do
        curl https://api.tria.ge/v0/search?query=sha256:$hashes >>
        /home/jarvis4444/Desktop/Triage_Output/Search/$hashes \

        --header 'Authorization:
        Bearer APIKEY'
done < hashes
```

Samples.sh

```
#!/bin/bash
while read sampleids; do
        curl https://api.tria.ge/v0/samples/$sampleids/overview.json >>
        /home/jarvis4444/Desktop/Triage_Output/Samples/$sampleids \

        --header 'Authorization:
        Bearer APIKEY'
done < sampleids
```

D.0.4 Linux

Move.sh

```
for i in $( ls ); do mv $i $i.json; done
```

jq.sh

```
#!/bin/bash
for f in *; do
        jq -n 'input' $f >newfile && mv newfile $f
done
```

D.0.5 YARA Rules

CosineSimilarity.py

```
import numpy as np
from numpy import dot
from numpy.linalg import norm

with open("yara.csv") as f:
    lines = f.readlines()
tags = []
apts = []
data = {}
for line in lines[1:]:
    s, a, t = line.strip().split(",")
    apts.append(a)
    tags.append(t)
    if a in data.keys():
        data[a]["tags"].append(t)
        data[a]["samples"].append(s)
    else:
        data[a] = {"tags": [t], "samples": [s]}
for d in data.keys():
    data[d]["cnt"] = len(list(set(data[d]["samples"])))
apts = list(set(apts))
tags = list(set(tags))
apts.sort()
tags.sort()
vectors = []
for a in apts:
    v = []
    for t in tags:
        v.append(round(data[a]["tags"].count(t) / data[a]["cnt"], 2))
    vectors.append({"apt": a, "vector": v})
correlations=[]
for i in range(len(vectors)):
```

```
    v1=vectors[i]
    for j in range(i+1,len(vectors)):
        v2=vectors[j]
        if v1["apt"] != v2["apt"]:
            a = v1["vector"]
            b = v2["vector"]
            correlations.append(["%s,%s"% (v1["apt"], v2["apt"]),
                  dot(a, b) / (norm(a) * norm(b))])
correlations.sort(key = lambda x: x[1])
for c in correlations[-100:]:
    print(c[0],round(c[1],2))
```