

The MITRE logo is displayed in a bold, white, sans-serif font. It is positioned in the top left corner of the page, above a large background image of a team working on computers in a dimly lit room. The image shows several people looking at a large screen displaying complex data and code, with a focus on a man in the foreground who is resting his chin on his hand, looking intently at the screen.

SOLVING PROBLEMS
FOR A SAFER WORLD®

ACTIVE DEFENSE CAPABILITY SET

TECHNICAL MANUAL

by Travis Gloor, Eric Hazard, Ronald Mercado, and Denise Olsen

APRIL 2022

ABSTRACT

This Technical Manual has been developed to support the United States European Command (USEUCOM) Joint Cyber Center Defensive Cyber Operations and the Cyberspace Theater Security Cooperation Office to strengthen the cyber posture of United States Allies and Partners in the region. The MITRE Corporation developed a four-phase approach to providing the Active Defense Capability Set (ADCS) Package to partner nations, and this technical manual is part of that capability set. The approach consists of the following phases:

1. Provide training on MITRE's ATT&CK™ for Enterprise Framework, Tactics, Techniques, and Procedures (TTP) Based Hunting and Operationalizing Cyber Threat Intelligence
2. Integrate a cyber hunt system with seven days of on-the-job training provided by MITRE on how to execute the 7-Step process outlined in this manual
3. Develop a technical manual providing a summary of how to execute the 7-Step TTP Base Hunt Methodology
4. Provide a Mission Qualification Certification Course to determine how well each cyber operator mastered the training that was provided.

The goal of the ADCS Package is to provide a repeatable process for developing partner nations to use on how to conduct TTP-based hunting with a strong emphasis of sharing cyber threat intelligence identified during execution of the 7 Steps. The United States as well as the European Union realizes that these adversaries know no boundaries. To successfully thwart cyber-attacks, we must jointly work towards educating our Partners and Allies by providing a repeatable process focused on defensive operations with the goal of enabling proactive defense to deter nation states by building partner capacity in cyber operations. Although this document was developed for USEUCOM Partner Nations, it is recommended that all Combatant Commands leverage this technical manual to strengthen partner coalition efforts in the cyber domain.

TO SUCCESSFULLY
THWART CYBER-
ATTACKS, WE MUST
JOINTLY WORK
TOWARDS EDUCATING
OUR PARTNERS AND
ALLIES BY PROVIDING A
REPEATABLE PROCESS
FOCUSED ON DEFENSIVE
OPERATIONS WITH THE
GOAL OF ENABLING
PROACTIVE DEFENSE TO
DETER NATION STATES
BY BUILDING PARTNER
CAPACITY IN CYBER
OPERATIONS.

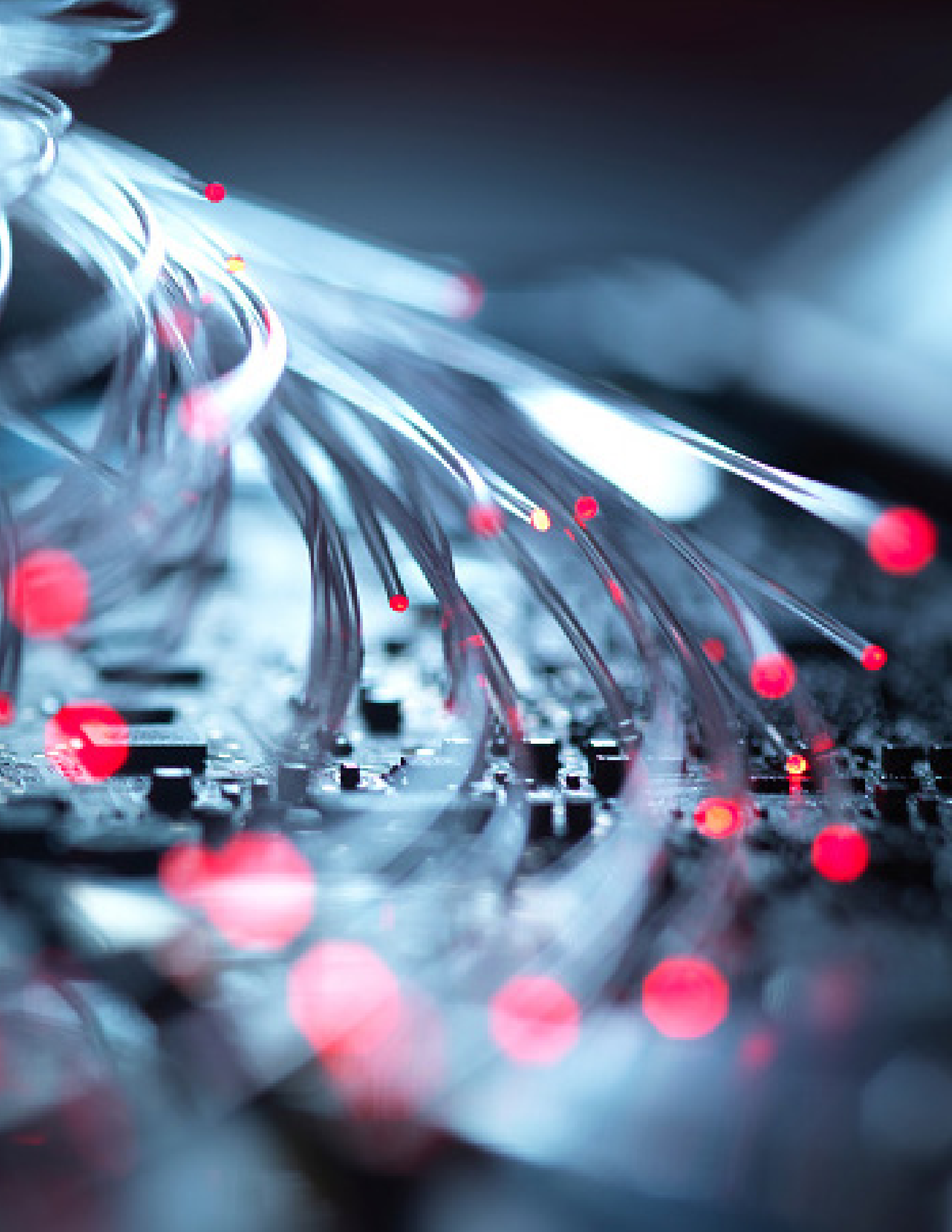
CONTENTS

ABSTRACT	I
CONTENTS	II
OVERVIEW	1
Goals and Objectives	2
Definition of Hunting	2
Prerequisite Training	3
Overview of TTP-Based Hunting	3
Pyramid of Pain	5
PREPARING FOR THE 7-STEP PROCESS	7
Active Defense Capability Set Tools	8
Deploy Sensors	10
STEP 1: DEVELOP A MALICIOUS ACTIVITY MODEL	15
Defining Malicious Activity	15
Creating a Malicious Activity Model	16
Leveraging Cyber Threat Intelligence	19
Using Computer Incident Response Center Luxembourg's Malware Information System Platform	21
MISP Objects	22
Searching in MISP	31
Mapping MISP Event Attributes to ATT&CK Framework	36
STEP 2: DEVELOP HYPOTHESES AND ABSTRACT ANALYTICS	39
Defining a Hypothesis and Resources to Inform Hypothesis and Analytic	39
Building the Hypothesis and Analytic via MITRE's CAR	40
Building the Hypothesis and Analytic via ATT&CK	43
Building the Cyber Hunt Plan	45

STEP 3: DETERMINE DATA REQUIREMENTS	46
An Overview of Data Types	46
Leveraging ATT&CK Data Sources	48
Determining Sysmon Data Requirements for T1053	50
Taking Advantage of Additional Resources	53
 STEP 4: FILTERING THE SOURCES OF DATA	 54
Information Required from the Network Owner	54
Understanding Time, Behavior, and Cyber Terrain	55
Using Nmap and Zeek to Begin to Filter	58
 STEP 5: IDENTIFY AND MITIGATE DATA COLLECTION GAPS	 60
Identifying Data Using Security Onion	60
Alerts	61
Hunt	63
PCAP	64
CyberChef	65
Grafana	66
TheHive	67
Suricata	
Strelka	68
Osquery and Fleet	68
Zeek	
Kibana	74
Relating Tools to the Cyber Hunt Plan	75
Improving Logging and Recommended Visibility of the Network	76
Filling Data Gaps by Deploying New Sensors	77

STEP 6: IMPLEMENT AND TEST ANALYTICS	79
Implementing Pseudocode Analytic to Kibana	79
Testing Analytics	80
Exploring Adversary Emulation	81
 STEP 7: HUNT/DETECT MALICIOUS ACTIVITY AND INVESTIGATE	 83
Locating Threat Hunting in the Incident Response Process	83
Tuning Analytic(s) for Initial Detection	86
Evaluating Events	88
Documenting Malicious Events	89
Gathering Contextual Information	91
Related Processes	91
Network Information	91
System Files	92
User Information	92
Investigating Malicious Events	93
Concluding the Cyber Hunt Plan	94
Responding to the Security Incident	97
Assess Analytics and Hunt Process	98
Additional Considerations	99

APPENDIXES	104
Appendix A: APT28 ATT&CK Techniques	105
Appendix B: APT28 & APT29 Compare Open Source	108
Appendix C: Categorized Tools	110
Appendix D: Bibliography	122
Appendix E: Training Resources	124
Appendix F: List of Figures	129
Appendix G: List of Tables	131
Appendix H: Abbreviations and Acronyms	132
 ABOUT THE AUTHORS	 134
 ABOUT MITRE	 135



OVERVIEW

This technical manual guides cyber operators, also known as cyber hunt teams, in executing a cyber hunt operation on any given network. To this end, the MITRE team supporting U.S. European Command surveyed tools used by cyber operators across the Department of Defense military services, identified over 105 tools of value to operations of interest for the users of this manual, and categorized those tools into 12 sets.

The categories and associated software tools are referenced in the Appendixes of this manual.

The team then performed a down-select of those tools per category and identified the top two or three tools as the best-of-breed, open-source tools. To target the most practical and easily used of those tools, the team developed a criteria-based approach to down-select further. Those criteria require that the software meets the characteristics below:

- Is available as open-source
- Offers a simple user experience, easy tool setup, and ease of tool configuration
- Offers additional functions and features to enhance user experience
- Includes documentation for installation, configuration, and use the tool
- Provides user support
- Is scalable and adapts easily to workload
- Is frequently updated
- Can be modified or developed to an end user's requirements
- Can be measured by confidence of reliability

The MITRE technical report, *TTP-Based Hunting* (Daszczyszak, Ellis, Luke, & Whitley, 2020) will be referenced throughout this manual as it is the baseline for the 7-step methodology that will be used. The report is linked in Training Resources in the Appendix.

THIS MANUAL IS
DESCRIPTIVE RATHER
THAN PRESCRIPTIVE.
OPERATORS MAY USE
ANY TOOL TO SUPPORT
THE METHODOLOGY.

Goals and Objectives

The goal of this manual has two facets:

- Deliver user-friendly technical guidance that describes leading-edge, industry best practice, and MITRE's tactics, techniques, and procedures (TTP)-based hunting methodology
- Supply users with a curated list of tools that cyber operators could use while executing the methodology

Under the goal, there are two objectives the manual sets out to accomplish:

- Provide a standardized repeatable approach to leveraging open-source tools that enable cyber operator to identify malign activity and/or adversary persistent threats on networks
- Curate sharable information on TTP-based hunting for the community of interest

This manual is descriptive rather than prescriptive. Operators may use any tool to support the methodology. Figure 3, maps the tools against the methodology's 7-step process. This manual is for cyber operators and/or analyst with over one year of experience in cyber operations.

Definition of Hunting

Throughout this manual, hunting is defined as “the proactive detection and investigation of malicious activity within a network” (Daszczyszak, Ellis, Luke, & Whitley, 2020). Threat hunting is a manual effort and human-centric process in proactive detection and organizations need to have time and the personnel dedicated to have an effective threat hunting program. To build an effective threat hunting team requires skilled personnel with different strengths at understanding different data types, strong researching skills, and an understanding of different adversarial techniques. Having skilled individuals that excel at understanding different data types with overlapping underlying skills will enable collaboration and understanding into what each member is investigating. For example, if a member of the threat hunt team is strong with network-based data, then adding other team members that are strong in host-based data or digital forensics will balance out the team. Hunting can be used interchangeably with the terms threat hunting, cyber hunt, and hunt operations. This document will use the terms operator, cyber threat hunters, threat hunters, and hunters interchangeably.

Prerequisite Training

Below is a recommended list of pre-requisite training to assist operators in successfully navigating through the TTP-based hunt 7-step process. In addition to the training operators may receive through their organization, no-cost training is available on the internet via product demonstrations and/or social media platforms such as YouTube. Additional training resources are listed in the Appendixes of this manual. Cyber operators will derive the most value from this document by reviewing the following resources:

- Elasticsearch, Logstash, and Kibana: <https://www.youtube.com/watch?v=v69kyU5XMFI>
- Suricata: https://www.youtube.com/results?search_query=training+on+suricata
- Nmap: https://www.youtube.com/results?search_query=training+on+nmap
- MITRE's Cyber Analytics Repository (CAR): https://www.youtube.com/results?search_query=mitre+cyber+analytics+repository
- The MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK)[™] Framework: https://www.youtube.com/watch?v=kQIISQ4XR_Q
- The MITRE ATT&CK Navigator: https://www.youtube.com/results?search_query=mitre+navigator
- The MITRE ATT&CK Defender: <https://mitre-engenuity.org/mad/>

Let's get started.

Overview of TTP-Based Hunting

In an effort to conceptually codify a method of threat hunting in the cyber domain, MITRE released *TTP-Based Hunting Methodology* (Daszczyszak, Ellis, Luke, & Whitley, 2020), which establishes a baseline for threat hunting and a 7-step methodology for conducting a TTP-based hunt. This baseline includes best practices for threat hunting, detection methods, and categorizing data. The 7-step methodology is most easily visualized as a “V” with two components: *characterization* of malicious activity and *hunt execution* (see Figure 1).

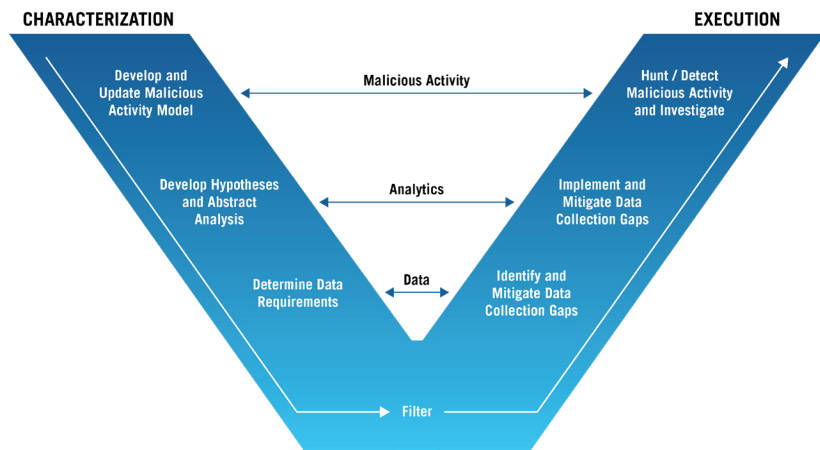


FIGURE 1. TTP-BASED HUNT METHODOLOGY “V” DIAGRAM

Daszczyszak, Ellis, Luke, & Whitley (2020) report the following:

Characterization of malicious activity starts with developing or updating the generic adversary model of behavior to identify all TTPs that an adversary may use—regardless of which adversary group, environment, or targeted network. For each TTP identified in the model, an analyst proposes one or more detection hypotheses that are formulated as abstract analytics. These hypotheses and abstract analytics are used to determine what data is necessary to collect. For each hunting operation, the hunt team should filter these data collection requirements and analytics based on the specifics of the terrain and situation of that hunt.

Execution employs the filtered data requirements and data model to conduct a gap analysis of sensors and data sources within the environment. If necessary, additional sensors (network or host-based) may be deployed at this stage to address visibility gaps. Once data is flowing into the analysis system, the analyst leverages the data model to implement analytics within the analysis system. The hunt team then executes the hunt by selecting specific analytics strongly associated with malicious behavior to try and obtain an initial detection. Analytic tuning and triaging suspicious and correlated events to positively identify the presence of an adversary follows this initial detection. (pp. 10-11)

This methodology focuses on TTP-based threat detection methods leveraging network-based and host-based data for anomaly detection and to identify Indicators of Compromise (IOCs). Each of these detection methods have benefits and limitations (Daszczyszak, Ellis, Luke, & Whitley, 2020, pp. 5-8). However, this manual is agnostic to the detection method and introduces tools that could instantiate any of the detection methods.

Pyramid of Pain

David Bianco's Pyramid of Pain (see Figure 2) visualizes attributes to detect on an adversary as a hierarchy. Each level of the pyramid corresponds to how difficult it is for the adversary to change those attributes or the level of pain that would cause an adversary to detect on those attributes. All levels should be considered during an engagement but the higher up the pyramid operators are developing content around the more effective the threat hunt engagement will be.

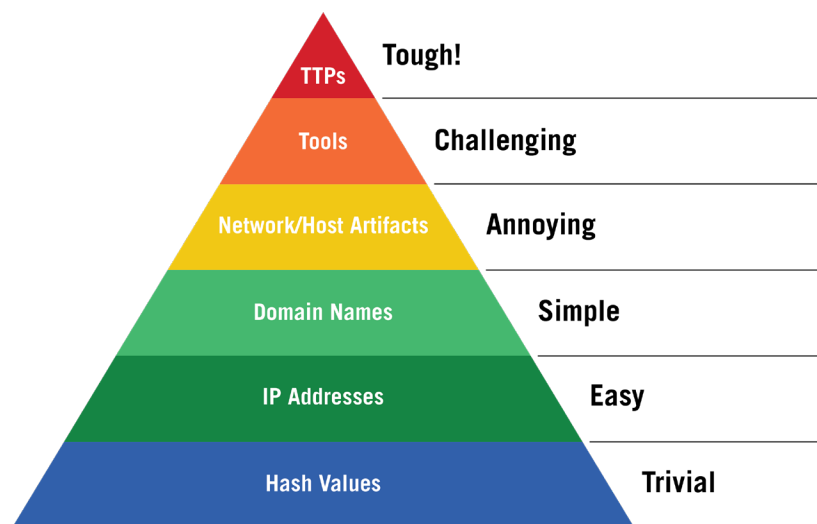


FIGURE 2. PYRAMID OF PAIN (BIANCO, 2014)

MITRE suggests that all security personnel take the time to understand every level of the pyramid (see Table 1 for definitions).

TABLE 1. PYRAMID OF PAIN IOCS

Indicators of Compromise	Value for Cyber Investigations
Tactics, Techniques, and Procedures (TTPs)	<i>TTPs</i> are the behaviors of a cyber actor. MITRE ATT&CK expresses TTPs concisely and thoroughly. For example, terms such as privilege escalation or initial access are complemented with corresponding techniques and procedures. Identifying and developing detections around adversary TTPs can be challenging but the level of pain is meant to articulate how difficult it is for adversaries to change those attributes. TTPs are also agnostic to the tools that the adversary might change or to the domains the adversary might choose to manipulate.
Tools	<i>Tools</i> are any given software or utility that an adversary has leveraged to achieve their goal (e.g., initial infiltration, obfuscation). These tools can be traditional tools found on computer systems (e.g., text editors) or malware built for malicious means (e.g., ransomware). If an adversary tool has been identified, operators can learn about the goal of the adversary and their approach.
Network or Host Artifacts	<i>Network or host artifacts</i> are any given item (e.g., dropped files, code typos) that an adversary might leave behind and thus signify an intrusion. These artifacts signal the adversary's trail; indicating where the adversary has been and what they touched.
Domain Names	<i>Domain names</i> are the identification string that represents "realms of authority" on the internet (e.g., google.com). As with known malicious Internet Protocol (IP) addresses, domain names (known malicious domains) can be used to find patterns in network traffic and search for anomalies.
IP Addresses	<i>IP addresses</i> are the assigned labels for any given computer connected to a network. These are of value to an investigation since IP addresses can be used to find patterns in a network or unwanted traffic from specific hosts.
Hash Values	<i>Hash values</i> are the output of hashing algorithms, such as Message Digest 5 (MD5) or Secure Hash Algorithm (SHA), designed to produce a unique fixed-length value for a piece of data. These values are helpful in signature-based detection since the function cannot be changed. While the hash of a given file has the potential to identify a malicious file, changing a single bit in a file can produce a different hash value enabling a malicious file to evade detection.

IOCs require context to provide real value to a threat hunt engagement since context helps answer the who, what, where, why, and how. These details help operators with initial detection, additional analysis upon discovery, and triaging alerts so that threat hunt teams aren't overwhelmed. However, IOCs are only known after the first attack is detected by some other means, analyzed, and shared with others to incorporate in their defenses before they are victimized. The time required for this process of analysis, sharing, and incorporation is longer than the time it takes an adversary to change IOCs low on the Pyramid of Pain. Therefore, there are strong limits on their usefulness. By the time IOCs are known, it is often "too late" (Bianco, 2014).

PREPARING FOR THE 7-STEP PROCESS

In preparation for the 7-step process, operators should become familiar with cyber related activities associated with Advanced Persistent Threat 29 (APT29) by conducting research on the internet. APT29 will be used as an example advanced persistent threat (APT) to model malicious activity. A Cyber Hunt Plan will be developed over the course of executing the 7-step process using discoverable data and/or information for this group. APT29 (also known as Cozy Bear) is a sophisticated cyber threat group that has targeted European and North Atlantic Treaty Organization (NATO) government networks and is commonly associated with the Russia's Foreign Intelligence Service (MITRE, 2017). The open-source community has published extensive research in APT29 TTPs which are referenced multiple times throughout this document. The Cyber Hunt Plan developed within this document is built around APT29 related activity.

In Figure 3 below, MITRE mapped the software tools to the 7-step methodology. The figure is a representation of the TTP-Based Hunt Methodology process in graphic form (Daszczyszak, Ellis, Luke, & Whitley, 2020).

IN PREPARATION
FOR THE 7-STEP
PROCESS, OPERATORS
SHOULD BECOME
FAMILIAR WITH CYBER
RELATED ACTIVITIES
ASSOCIATED WITH
ADVANCED PERSISTENT
THREAT 29 (APT29)
BY CONDUCTING
RESEARCH ON THE
INTERNET.

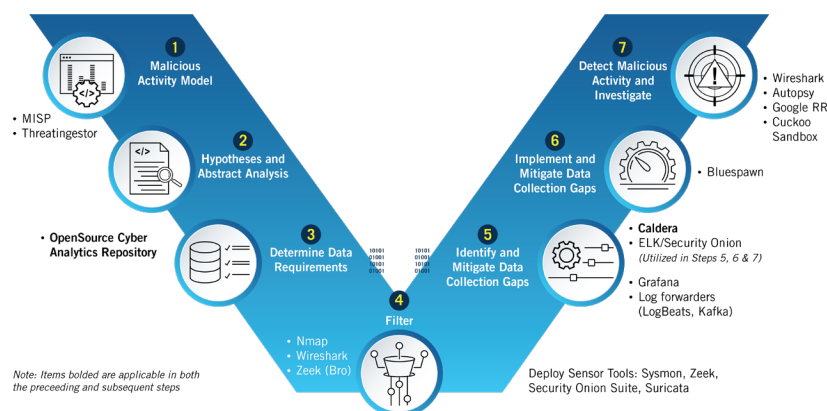


FIGURE 3. TTP-BASED HUNT METHODOLOGY, WITH MAPPED TOOLS

Active Defense Capability Set Tools

Cyber operators are only as effective as the tools they employ and tune to their specific use case. This recommended set of Active Defense Capability Set (ADCS) tools will allow operators to collect and analyze data to potentially detect anomalous or malicious activity. Below are several tools that are used in the ADCS solution to aid operators to execute cyber hunt activities that are explored within this document. Additionally, a summary of all the tools will also be provided in the Appendixes within this document.

1. Security Onion: Security Onion is a free and open Linux distribution for threat hunting, enterprise security monitoring, and log management. The Linux distribution includes Playbook, Elasticsearch, Logstash, Kibana, Suricata, Zeek, and many other security tools. It is the underlying operating system supporting the tools to generate data and metadata, detect on the collected data, index and, normalize the data, and allow operators to query and analyze the collected data (Security Onion, 2021). <https://docs.securityonion.net/en/2.3/about.html#security-onion>
2. Zeek: Zeek is a passive, open-source network traffic analyzer. Many operators use Zeek as a Network Security Monitor (NSM) to support investigations of suspicious or malicious activity. Zeek generates an extensive set of logs describing network activity. These logs include not only a comprehensive record of every connection seen on the wire but also application-layer transcripts (Zeek, 2021). <https://docs.zeek.org/en/master/about.html>
3. Suricata: Suricata is a high-performance Intrusion Detection System (IDS), Intrusion Prevention System (IPS), and NSM engine. It is open-source and owned by a community-run non-profit foundation, the Open Information Security Foundation (OISF). Suricata inspects the network traffic using a powerful and extensive rules and signature language and has powerful Lua scripting support for detection of complex threats (Suricata, 2019). <https://suricata.readthedocs.io/en/suricata-6.0.3/what-is-suricata.html> <https://docs.securityonion.net/en/2.3/suricata.html>
4. Elasticsearch, Logstash, Kibana (ELK): The ELK stack is a collection of open-source products that, together, allow organizations to employ a centralized Security Information and Event Management solution. Elasticsearch is a full-text and analysis engine. Logstash is a log aggregator capable of collecting data from various data sources,

can execute different transformations and enhancements, and then transfers that data to various supports destinations. Kibana is a data analytics tools that enables operators to analyze and visualize the data (Horovits, 2020). <https://logz.io/learn/complete-guide-elk-stack/>

5. System Monitor (Sysmon): Sysmon is an open-source, Windows system service and device driver that, once installed on a Windows system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time (Russeinovich & Garnier, Sysmon v13.24, 2021). <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
6. Malware Information Sharing Program (MISP): MISP (Open-Source Threat Intelligence and Sharing Platform) software facilitates the exchange and sharing of threat intelligence, Indicators of Compromise (IOCs) about targeted malware and attacks, financial fraud, or any intelligence within the community of trusted members. MISP sharing is a distributed model containing technical and non-technical information which can be shared within closed, semi-private or open communities. Exchanging such information should result in faster detection of targeted attacks and improve the detection ratio, whilst also reducing the number of false positives (CIRCL, 2021). <https://github.com/MISP/MISP>
7. Stenographer: Stenographer is a full-packet-capture utility for buffering packets to disk for intrusion detection and incident response purposes. It provides a high-performance implementation of Network Interface Controller (NIC) to disk packet writing, handles deleting those files as disk fills up, and provides methods for reading back specific sets of packets quickly and easily. Used in the latest Security Onion to facilitate collection of Packet Capture (PCAP) (Google, 2020). <https://github.com/google/stenographer>
8. Network Mapper (Nmap): Nmap is a free and open-source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime (NMAP, n.d.). <https://nmap.org>

For the purposes of this manual, a *sensor* is defined as something that collects information about a network and can be used to make decisions about the network's security. Sensors can be grouped into network-based sensors and host-based sensors. *Network-based sensors* include tools like Zeek, Suricata, and PCAP collection. *Host-based sensors* include Windows Event Logs, Sysmon, osquery, and Wuzuh. The term security sensor refers to the hardware used to collect and analyze the data generated from the network-based and host-based sensors. Effective monitoring of a network builds on data collected from multiple sensors. Collecting data from host-based sensors provides a wider detection range for identifying different TTPs compared to network-based sensor data. Some TTPs can be detected using either data type but certain TTPs are dependent on one or the other data types for detection. Figure 4 shows the detection capabilities of the two data types in relation to the MITRE ATT&CK™ (Adversarial Tactics, Techniques, and Common Knowledge) framework.



It's good practice to identify sensors according to the type of data each can collect. Operators will also want to identify their placement in the network based on locations that will provide the maximum visibility required to collect data. Data sources are covered in Step 3: Leveraging ATT&CK Data Sources on page 46.

While deploying sensors is crucial for security operations and threat hunting operations, operators and administrators need to be cognizant of the potential impact to the network and systems when introducing sensors to the infrastructure. Forwarding host-based data will have an impact on network bandwidth as that data traverses the network. MITRE recommends testing on a subset of systems to determine new baselines, and a tiered distribution should be considered. Deploying sensors is dependent on the situation per environment, and a rapid deployment may be the only option. Operators also need to consider the amount of memory that is going to be required on the security sensors to ingest, normalize, visualize, and retain the collected data.

Network-based sensors copy the data on the network segment through one of two methods: passive and in-line. Passive deployments require the use of a Test Access Port (TAP) or Switchport Analyzer (SPAN) to be configured on a networking device to copy the data on the network segment and send that copied data to the security sensing solution. In-line deployments place the security sensor on the network segment and the data being monitored must travel through the security sensor to be collected. The security sensor must have a promiscuous port configured to receive the collected network-based traffic, which means the interface on the hardware will collect all the traffic being sent to it instead of just the traffic the controller is specifically meant to receive. In order to change the interface into a promiscuous port, the system administrator must adjust the settings on the hardware itself.

Passive deployments are preferred as most aggregation TAPs use fail-open concepts allowing traffic to flow through the network interfaces on the TAP preventing traffic from being disrupted. Host-based sensors generate the data on the host and must be forwarded to the security sensor for analysis or viewed on the endpoint that generated the events. Being able to forward the host-based data to a single location allows operators to aggregate the results to identify anomalous behavior and trends across multiple hosts at once. Vendors for data analytic tools provide their own forwarding agents and documentation on the procedures to forward data to its platform. The Elasticsearch Beats forwarding agent is used within ADCS since ADCS leverages the ELK stack.

**MITRE RECOMMENDS
TESTING ON A SUBSET
OF SYSTEMS TO
DETERMINE NEW
BASELINES, AND
A TIERED DISTRIBUTION
SHOULD BE
CONSIDERED.**

Identifying when to apply these methods and where to monitor on the network is key to collecting appropriate data for threat hunting. Refer to Figure 5 to show how network data is copied to the sensor's promiscuous port.

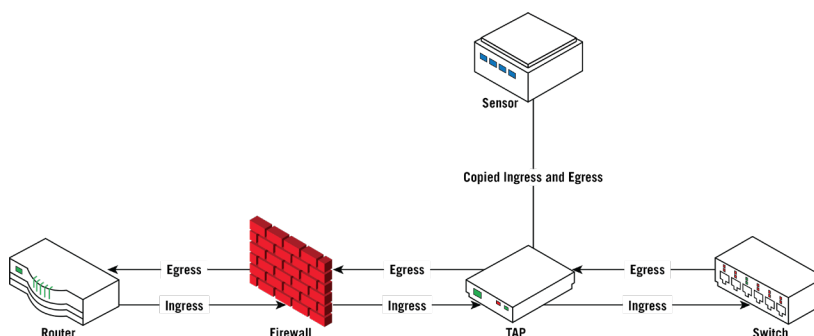


FIGURE 5. SIMPLIFIED NETWORK TAP PLACEMENT

Figure 5 shows Ingress and Egress traffic passing through the TAP and making an exact network copy for a sensor, such as Zeek and Suricata to process this information and develop analytics of the network behavior. Throughput of a NIC tells how much data can transfer through the interface, for example a 1 gigabyte (Gb) NIC can pass up to a gigabyte per second of data a second. When deploying a TAP, ensure that the TAP NIC's throughput matches the device's NIC throughput to keep the TAP from limiting the network connection, affecting resources, and causing possible network latency. Powered TAPs with redundant power supplies are preferred instead of power over ethernet because powered TAPs will not draw current from the network, which would affect signal strength.

Another network monitoring method is using a SPAN on a networking device that has SPAN built in. SPAN duplicates network packets passing through the selected ports and passes them out a specified port to the security sensor. The benefit of using a SPAN is that it reduces the impact to the security budget since no extra hardware is needed. The disadvantage is that SPANs are reliant on the networking devices resources of ports, memory, and bandwidth. Before setting up a SPAN, make sure the networking device utilization of ports is considered. A 1 Gb, 12-port switch (with each port using a full Gb of traffic) will produce more data than the SPAN port can capture, and this will lead to a loss of data. SPAN ports should be used in low-traffic parts of the network. SPAN traffic is

considered low priority data on a switch, meaning if the switches resources are at high utilization rates then the SPAN traffic will be dropped before the switched network traffic.

In high-traffic areas on the network, TAPs are preferred over SPAN ports, as the TAP can provide a more reliable representation of the network traffic. The most effective location for a TAP is between the router and switch. A common practice to help maximize network traffic data collection is to use the TAP and SPAN together, with the TAP in high-traffic volume areas and SPAN in low-traffic area. This will expand the visibility of data flow and monitor more of the network. Figure 6 shows a more advanced network architecture using multiple TAPs.

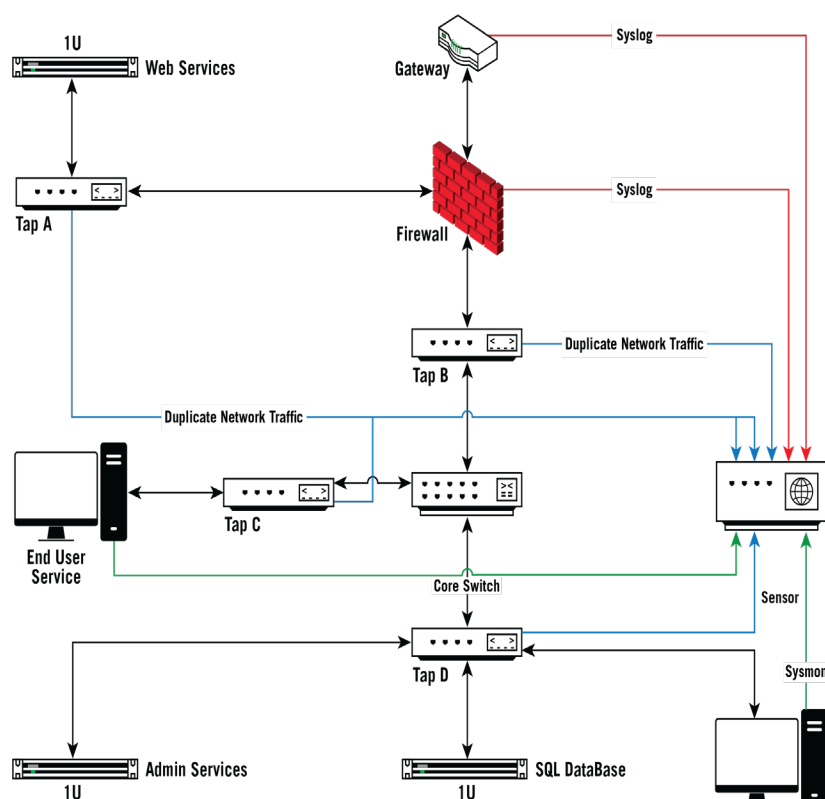


FIGURE 6. EXPANDED EXAMPLE OF NETWORK TAP PLACEMENTS

When the Admin Service system communicates with an End-User Device, this is lateral movement. TAP A and B will not record this traffic due to not being on the network path for this communication. Depending on the complexity of the network, tapping in multiple locations is needed to monitor the full network. The more TAPs on the network, the more resources that will be consumed on the sensor for processing the increase in network data. This is where a Networking Monitoring tool like Nmap is useful to help map out the network and ensure that the entire network is accounted for. This will help with detecting gaps in the networks security monitoring that will be discussed later in this document.

In Figure 6 on the previous page, there is lateral communication that is not monitored travelling between the Admin Service and the Structure Query Language (SQL) Database. If the communication between these systems is determined to be a low-traffic area, then a SPAN port can monitor and send the duplicated network traffic to the sensor. This would allow for the entire network to be monitored using both TAP and SPAN.

STEP 1: DEVELOP A MALICIOUS ACTIVITY MODEL

Defining Malicious Activity

Malicious cyber activity refers to actions that are not authorized by nor in accordance with law and that seek to compromise or impair the confidentiality, integrity, or availability of (a) computers information or communication systems; (b) network, physical, or virtual infrastructure controlled by computers or information systems; or (c) information resident thereon (NIST, n.d.). The purpose of Step 1 is to begin identifying the signs of known and unknown malware, malicious use of legitimate tools, and zero-day exploits that cyber adversaries use to get an initial foothold in the network environment so that operators can identify the adversary and associated tactics, techniques, and procedures (TTPs). Adversaries can use malicious code (e.g., computer viruses, worms, trojans, spyware, logic bombs, adware, and backdoor programs) to execute malicious activity. A malicious code is a harmful computer programming script designed to create or exploit system vulnerabilities.

Below is a list of several examples of malicious activity (MITRE, 2021):

- **Social Engineering:** The manipulation of people into performing actions or divulging confidential information (e.g., username and password for computer accounts) to perform malicious activity.
- **Phishing:** A type of social engineering attack often used to steal user data, including login credentials and credit card numbers. Occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message.
- **Spear Phishing:** An electronic communication, usually an email directly targeting a specific organization, group, or specific individuals.
- **Man in the Middle:** When a cyber actor intercepts communications between two parties either to secretly eavesdrop on or to modify the communications traffic traveling between both parties.
- **Data Encrypted for Impact:** Adversaries may encrypt data on target systems in a network to interrupt availability of systems and network resources. Most commonly seen in Ransomware, adversaries use the technique for monetary compensation from the victim or to render data permanently inaccessible.
- **Denial of Service:** A malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming or flooding the system with traffic to the point that the system crashes or is unavailable to its intended users.

STEP 1 COVERS

- Defining malicious activity
- Creating a malicious activity model and provide an overview of the Diamond Model (Figure 7)
- Leveraging Cyber Threat Intelligence (CTI) to support the model
- Introducing the use of MISP, which MITRE recommends threat hunt teams use to gather information on malicious activity found by other organizations and users
- Providing highlights on searching in MISP to support the development of a malicious activity model
- Mapping MISP events and how they correlate to the MITRE ATT&CK Framework

AS THREAT HUNTING PROGRAMS MATURE, PROVIDING A FRAMEWORK OR MALICIOUS ACTIVITY MODEL CAN HELP STRUCTURE THE THREAT HUNTING PROCESS.

Creating a Malicious Activity Model

Many organizations are moving to TTP-based threat detection to enable their detection engineering and threat hunting operations while the current standard is to leverage open-source intelligence on the threat actors or APT groups that are most applicable to their industry. Both methods are effective and should be employed in security operations. Once those groups are identified, performing research on published artifacts, IOCs, and extracting actionable intelligence from narrative reports can drive threat hunting operations. As threat hunting programs mature, providing a framework or malicious activity model can help structure the threat hunting process. This document uses the Diamond Model of Intrusion Analysis (Caltagirone, Pendergast, & Betz, 2013) (see Figure 7), which serves as an initial framework for asking questions to identify potential adversaries and malware. The Diamond Model shows how an adversary uses a capability over a particular infrastructure against a victim. The core features of the model include adversary, capability, infrastructure, and victim. The adversary feature is used to describe the potential threat to an organization's network.

- Typically, the adversary is an activity group name, such as APT29 (i.e., Cozy Bear).
- The capability feature describes the adversary's collection malware, tools, and TTPs. Additional groups can be identified on the MITRE ATT&CK website under groups. MITRE ATT&CK also has information about groups including their Identification (ID), associated groups, descriptions, identified TTP's, software used, navigator layers, and references to threat intelligence reports which can be found at <https://attack.mitre.org/groups/>.
- The victim feature describes the target (e.g., individual, company) in terms of industry, sector, personnel targeted, assets targeted, vulnerabilities, and country.
- The infrastructure feature describes the adversary's IP addresses, domain names, information about the hosting provider, Whois lookup details, and if the infrastructure is being used to anonymize the source of the activity.

Note: Attribution and the naming of activity groups can sometimes be confusing. Threat intelligence vendors and other groups performing CTI typically have their own criteria for which data points they use to develop an activity group.

The data points that each vendor can answer are based on their level of visibility into the activity in question. Vendors have visibility across multiple networks and other resources to collect information about multiple events. They add the information they collect to a known activity group or use it to develop a new group.

It's a good practice to treat activity groups based on vendor names as separate groups until the threat hunt team has confidence that the groups are associated. Perform an analysis by comparing the different groups and identify what the groups have in common. If operators are confident that the two groups are the same activity, then it's appropriate to treat them as the same activity group (MITRE, n.d.).

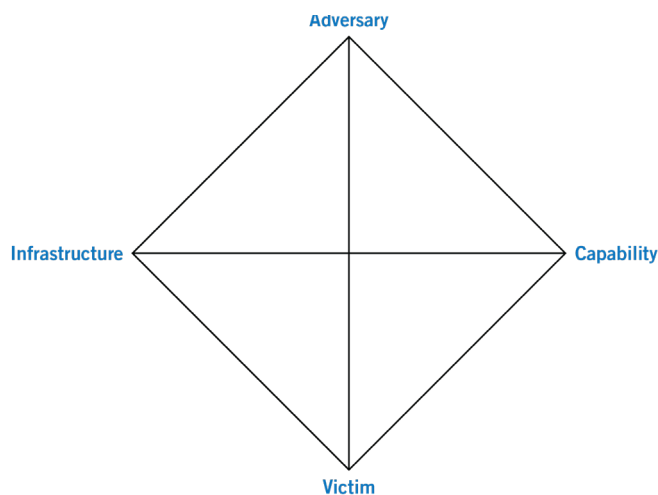


FIGURE 7. THE DIAMOND MODEL

With the Diamond Model, operators can use one of the features as a starting point and pivot to the other features. For example, starting with the *Victim* feature, operators can identify the characteristics of the potential network or victim that the threat hunt team will be supporting with an investigation. Threat hunt teams may be hunting on a network that is operated by a government organization such as a Ministry of Defense in a given country. Operators can then pivot to the *Adversary* by asking which groups have the intent and motivations to target a Ministry of Defense in that specific country or geographic region. Multiple adversary groups that could be potential threats to the organization may be identified.

Next, operators might prioritize the groups by the likelihood that the identified group could be actively targeting the organization that the threat hunting team will be hunting on. Starting with the highest priority adversary, operators can pivot to the *Capability or Infrastructure* features. For hunting, it's more useful to pivot to the capability feature to start building a list of possible malware and TTPs used by the identified adversary. From the Capability feature, operators can pivot to Infrastructure to identify known IP addresses and domain names that are associated with a given capability to provide context to those types of IOCs if those artifacts are identified during a hunting engagement (Caltagirone, Pendergast, & Betz, 2013).

Once the adversary has been identified and the operators have built a list of possible TTPs, they can start developing a model based around those TTPs. With the TTP's identified, cyber intelligence and operations become fused together at this point as operators can start developing detection techniques and queries to identify the behaviors. As the activity model is developed, operators should use a collaborative tool, such as a wiki or a shared spreadsheet to capture shared knowledge and update that tool as they work through the process.

Table 2 captures techniques and context together that will be used to develop a hypothesis in Step 2. If a technique has been identified that is associated with another technique or malware, the malicious activity model may be updated for additional context and/or be used for further investigation or actions.

TABLE 2. SAMPLE MALICIOUS ACTIVITY MODEL FOR APT29

APT29 (Associated Groups: YTTIRIUM and Cozy Bear)	
Technique	Context
T1053.005 Scheduled Task/ Job: Scheduled Task	APT29 used named and hijacked scheduled tasks to establish persistence.
T1218.011 Signed Binary Proxy Execution: Rundll32	APT29 has used rundll32.exe for execution.
T1047 Windows Management Instrumentation (WMI)	APT29 used WMI to steal credentials and execute backdoors at a future time.

Table 2 is an example of an adversary-based activity model. Depending on use case, operators may decide to develop an activity another way. Alternative activity models may be based on malware, common techniques associated with popular ransomware, or the most seen techniques for targeting a

particular sector, industry, or country. For example, if an organization feels they're susceptible to malware performing credential dumping LSASS memory the activity model could focus on Mimikatz, and they could develop a table focused on T1003.001 OS Credential Dumping: LSASS. There are several artifacts that can be used from Open-Source Intelligence (OSINT) that the activity model can focus on such as process command arguments and registry keys accessed. Operators should also incorporate reporting from internal investigations and incidents for techniques that went undetected by the operator's organization or the organization the threat hunt team is investigating on behalf of. In Table 3, a Cyber Hunt Plan is beginning to take shape using the referenced technique. As the methodology progresses, additional information will be added to the plan to use for the development of all the steps in the TTP-Based Hunt Methodology.

TABLE 3. CYBER HUNT PLAN

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.

Threat hunt teams can use the MITRE ATT&CK website as a starting point for known techniques associated with a group or software, but it should be supplemented with additional intelligence sources. Many publicly available intelligence reports and blogs map observed activities to the ATT&CK framework.

Leveraging Cyber Threat Intelligence

Cyber Threat Intelligence (CTI) is knowledge about the intentions, motivation, and methods of a cyber adversary respective to their tactics. CTI refers to sources of information (or actionable knowledge) about threats (malicious activity) and threat actors that help mitigate harmful events in cyberspace; this information may be obtained from open-sources, social media, humans, technology, or the deep/dark web (FireEye, n.d.). The data from CTI can help teams understand the who, what, where, when, how, and why questions regarding the threats. MITRE references CTI from the Malware Information Sharing Platform (MISP) as an example throughout this manual.

Additional CTI resources that are useful for collecting additional information for the malicious activity models that are developed are listed below:

- **Alienvault:** <https://otx.alienvault.com/>
- **CERT-EU:** <https://cert.europa.eu/cert/filteredition/en/CERT-LatestNews.html>
- **CrowdStrike Blog:** <https://www.crowdstrike.com/blog/>
- **Cybersecurity & Infrastructure Security Agency:** <https://us-cert.cisa.gov>
- **ESET welivesecurity Blog:** <https://www.welivesecurity.com/>
- **FireEye Blog:** <https://www.fireeye.com/blog.html>
- **JPCERT:** <https://blogs.jpCERT.or.jp/en>
- **Malpedia:** <https://malpedia.caad.fkie.fraunhofer.de/actors>
- **Microsoft Security Intelligence:** <https://www.microsoft.com/security/blog/microsoft-security-intelligence/>
- **OpenCTI:** <https://www.opencti.io/en/>
- **Recorded Future:** <https://www.recordedfuture.com/blog/>
- **Red Canary Blog:** <https://redcanary.com/blog/>
- **Securelist:** <https://securelist.com/>
- **SecureWorks Blog:** <https://www.secureworks.com/blog>
- **Symantec Blog:** <https://symantec-enterprise-blogs.security.com/blogs/>
- **Talos Blog:** <https://blog.talosintelligence.com/>
- **ThaiCert Threat Actor Encyclopedia:** <https://apt.thaicert.or.th/cgi-bin/aptggroups.cgi>
- **The Digital Forensic and Incident Response (DFIR) Report:** <https://thedfirreport.com/>
- **ThreatMiner:** <https://www.threatminer.org/>
- **Unit42:** <https://unit42.paloaltonetworks.com/>

Using Computer Incident Response Center Luxembourg's Malware Information System Platform

As cyber threats across the world become more sophisticated, targeted, widespread, and increasingly undetected, MITRE recommends that threat hunt teams join a community of interest (COI) to learn as much as possible about potential threats starting with MISP (MISP, n.d.). Developed by Computer Incident Response Center Luxembourg (CIRCL), MISP is an open-source, community-driven threat intelligence tool that enables sharing and storing both technical and non-technical information about malware samples, cyber adversaries, and specific cyber incidents or events. Since it is a community-driven tool, it enables members who do not have expertise to connect to a larger community of organizations that have experienced cyber threat analysts on staff. Information from MISP will be useful to operations and will help develop or refine the malicious activity model. General information on MISP can be found at <https://www.misp-project.org/index.html>.

Among the many MISP users are malware reverse engineers, security analysts, intelligence analysts, law enforcement supporting cyber investigations, and fraud and risk analysts (CIRCL Luxembourg, 2018). By becoming part of the COI within the MISP sharing platform, teams learn to be more proactive than reactive in finding cyber adversaries. Although MITRE encourages cyber operators to focus on tactics, techniques, and procedures at the top of the Pyramid of Pain, it is also encouraged that organizations become members of this COI since the contributors help other users gather information about cyber campaigns and cyber threat actors who may be targeting specific organizations or government sectors.

Operators can request access by contacting <https://www.circl.lu/contact/> or by downloading, installing, and creating a username and password for the organization's MISP instance. Once an account has been created and are connected to a live internet-facing network, operators are essentially now part of the MISP COI. Download the software here: <https://www.misp-project.org/download/>. Before accessing MISP, consider training on how to use it. MISP User Training Modules can be found at <https://www.misp-project.org/misp-training/1-misp-usage.pdf>.

INFORMATION FROM
MISP WILL BE USEFUL
TO OPERATIONS AND
WILL HELP DEVELOP OR
REFINE THE MALICIOUS
ACTIVITY MODEL.

Figure 8 displays the log in page.



FIGURE 8. LOG IN PAGE

MISP comes with prepopulated threat feeds once connected via the internet. Additional (licensed or open-source) feeds can be used such as Department of Homeland Security Automated Indicator Sharing, Federal Bureau of Investigations InfraGard, @abuse.ch Ransomware Tracker, Circl.lu, etc., to augment the sources. Once the intelligence feeds have been ingested, the data can be sent to tools such as Zeek, which is discussed in more detail in Step 5: Identifying Data Using Security Onion on page 60.

MISP Objects

In this section, high-level concepts of MISP Objects, creating Events and associating Attributes to Events, and their relationship to the MITRE ATT&CK Framework will be covered.

One of the first things operators need to master is the concept of objects. In MISP, objects detect, block, or perform intelligence gathering about campaigns and cyber-attacks. MISP objects are an advanced method of sharing combinations of attributes that are contributed by MISP users. The objects are dynamic because they can be used by other information-sharing platforms and enable real-time updates in operational distributed-sharing systems (important since security threats and indicators are also dynamic). Standard objects are static and incorporating new threat indicators requires significant time (Ikclody, Alexandre, & CIRCL, 2018).

These objects and their associated attributes are based on real cybersecurity use cases and existing practices in information sharing (MISP, 2021). As a MISP user, operators can develop and propose their own MISP objects and contribute their expertise to the COI.

New MISP users can find existing MISP objects from the following sources:

- <https://github.com/MISP/misp-objects/blob/main/README.md>
- <https://www.misp-project.org/objects.html>

A PDF file named “MISP Objects” identifies over 150 objects that can be used. These objects have been contributed by other MISP users or other intel-sharing platforms. Operators should read through the README.md file on github to learn about the defined values of *misp-attribute*, *ui-priority*, *the field of multiple values_list*, and other similar taxonomies. An object is described in a simple JavaScript Object Notation (JSON) file containing the following elements:

- **Name:** The name of an object
- **Meta-category:** The category where the object falls into (file, network, financial, misch, internal)
- **Description:** A summary of the object description
- **Version:** The version number as a decimal value
- **Required:** An array containing the minimal required attributes to describe the object
- **RequiredOneOf:** An array containing the attribute where at least one need to be present to describe the object
- **Attributes:** Another JSON object listing all the attributes composing the object

JSON is an open standard file format, as well as a data interchange format, which uses human-readable text to store and transmit data objects consisting of attribute-value pairs. An *attribute-value pair* is defined in JSON objects as *key/value pairs*. Keys must be strings, and values must be a valid JSON data type (string, number, object, array, Boolean, or null). Keys and values are separated by a colon. Each key/value pair is separated by a comma. An *array data type* is the minimal required attributes to describe the object (JSON, n.d.).

JSON IS AN OPEN
STANDARD FILE
FORMAT, AS WELL AS
A DATA INTERCHANGE
FORMAT, WHICH USES
HUMAN-READABLE
TEXT TO STORE AND
TRANSMIT DATA
OBJECTS CONSISTING
OF ATTRIBUTE-VALUE
PAIRS.

Figure 9 displays a state-of-the-art MISP data model with the associated information (CIRCL Team MISP Project, n.d.):

- **Galaxy:** A group of threat information or way to express a large object (called cluster) that can be attached to MISP events or attributes
- **Events:** There can be many associated with a Galaxy; events include the date, event information, threat level and source of the event
- **Objects:** An advanced method of sharing combinations of attributes
- **Tags:** The data elements associated with the Event or Attributes to enhance searching in MISP
- **Taxonomies:** A set of predefined classifications by Computer Security Incident Response Teams (CSIRT)/Computer Emergency Response Teams (CERT) or national threat classifications
- **Discussion:** Free text area; open discussion points may be inserted by individual who created the Event
- **Correlation Proposal:** Data associated from an Event or Attribute with an associated correlation

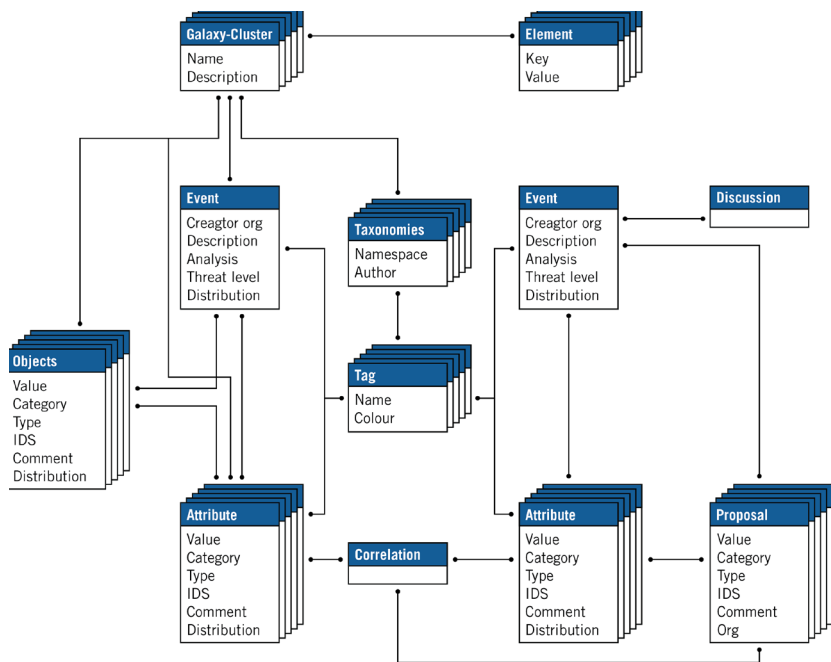


FIGURE 9. THE STATE OF THE ART MISP DATA MODEL

Figure 10 below, shows examples of a generic MISP Event Object with correlating attributes:

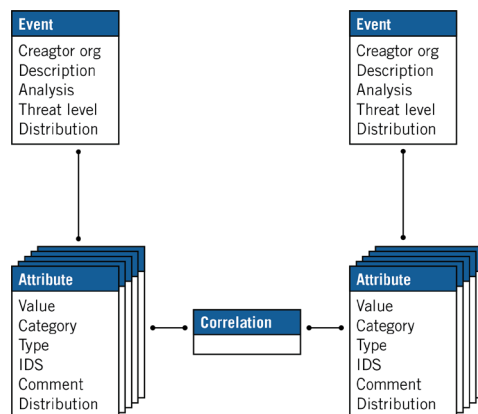


FIGURE 10. MISP EVENT OBJECT WITH ATTRIBUTES

The model is also helpful for showing how data is being shared among other MISP communities. According to CIRCL, more than 1,250 organizations and more than 3,600 users contribute to this model.

Figure 11 on the following page, displays an example of a MISP object template in JSON with a “Domain Attribute” (MISP, 2021). This example shows the information contained in the object template and how that information is categorized.

```

{
  "attributes": {
    "domain": {
      "categories": [
        "Network activity",
        "External analysis"
      ],
      "description": "Domain name",
      "misp-attribute": "domain",
      "multiple": true,
      "ui-priority": 1
    },
    "first-seen": {
      "description": "First time the tuple has been seen",
      "disable_correlation": true,
      "misp-attribute": "datetime",
      "ui-priority": 0
    },
    "ip": {
      "categories": [
        "Network activity",
        "External analysis"
      ],
      "description": "IP Address",
      "misp-attribute": "ip-dst",
      "multiple": true,
      "ui-priority": 1
    },
    "last-seen": {
      "description": "Last time the tuple has been seen",
      "disable_correlation": true,
      "misp-attribute": "datetime",
      "ui-priority": 0
    },
    "port": {
      "categories": [
        "Network activity",
        "External analysis"
      ],
    },
  }
}

```

FIGURE 11. EXAMPLE OF A MISP OBJECT TEMPLATE

MISP comes preloaded with threat information. This data is shown in specific galaxies. As an example, galaxy data types (or groupings) are identified as Threat Actor Galaxy, Malware Galaxy, etc. Within the MISP application, there are 43 pre-defined galaxies by data type. As mentioned earlier, galaxies in MISP are a method to express clusters (i.e., large objects) that can be attached to MISP events or attributes. A cluster is composed of one or more elements. Elements are expressed as key-values (MISP, n.d.). The Threat Actor Galaxy is important since it can search based on this particular galaxy, which will help develop or refine the malicious activity model. More information on MISP galaxy clusters can be found at www.misp-project.org/galaxy.html.

Figure 12 displays sample galaxies.

THE THREAT ACTOR
GALAXY IS IMPORTANT
SINCE IT CAN SEARCH
BASED ON THIS
PARTICULAR GALAXY,
WHICH WILL HELP
DEVELOP OR REFINE
THE MALICIOUS
ACTIVITY MODEL.

ID	Icon	Name	version	Namespace	Description	Actions
1	Android	Android	3	misp	Android malware galaxy based on multiple open sources.	edit delete
2	attackfraud	attackfraud	1	misp	attackfraud - Principles of MITRE ATT&ACK in the fraud domain	edit delete
3	Backdoor	Backdoor	1	misp	Malware Backdoor galaxy.	edit delete
4	Banker	Banker	3	misp	Banking malware galaxy.	edit delete
5	Botnet	Botnet	2	misp	Botnet galaxy.	edit delete
6	Branded Vulnerability	Branded Vulnerability	2	misp	List of known vulnerabilities and exploits	edit delete
7	Cert EU GovSector	Cert EU GovSector	2	misp	Cert EU GovSector	edit delete
8	Country	Country	1	misp	Country meta information based on the database provided by geonames.org.	edit delete
9	Election guidelines	Election guidelines	1	misp	Universal Development and Security Guidelines as Applicable to Election Technology.	edit delete
10	Exploit-Kit	Exploit-Kit	4	misp	Exploit-Kit is an enumeration of some exploitation kits used by adversaries. The list includes document, browser and router exploit kits. It is not meant to be totally exhaustive but aim at covering the most seen in the past 5 years.	edit delete
11	Malpedia	Malpedia	1	misp	Malware galaxy based on Malpedia archive.	edit delete
12	Microsoft Activity Group actor	Microsoft Activity Group actor	3	misp	Activity groups as described by Microsoft	edit delete
13	Malinformation Pattern	Malinformation Pattern	4	malinfosec	AMTT Tactic	edit delete
14	Attack Pattern	Attack Pattern	8	mitre-attack	ATT&ACK Tactic	edit delete
15	Course of Action	Course of Action	7	mitre-attack	ATT&ACK Mitigation	edit delete
16	Enterprise Attack - Attack Pattern	Enterprise Attack - Attack Pattern	5	deprecated	ATT&ACK Tactic	edit delete
17	Enterprise Attack - Course of Action	Enterprise Attack - Course of Action	5	deprecated	ATT&ACK Mitigation	edit delete
18	Enterprise Attack - Intrusion Set	Enterprise Attack - Intrusion Set	5	deprecated	Name of ATT&ACK Group	edit delete

FIGURE 12. EXAMPLE GALAXIES IN MISP

New galaxy data is challenging to add or change. New threat actors, malware, or campaigns can be added via the graphical user interface (GUI), but operators must have administrator-level access to the server hosting the MISP instance to modify or add additional galaxy data.

To see the list of events already available in the MISP instance and their related attributes, click the List Events tab on the right-hand side. The attributes, published organization, date, and distribution are listed (see Figure 13 on the following page).

Name	Org	Clusters	Type	Date	Info	Actions
1153	ame	1153		2021-01-05	trying to figure out embargo/leakage	Community < Not published
1152	ame	1152	Threat Actor	2020-07-20	2020-07-20	Community < Not published
1151	ame	1151	Threat Actor	2020-07-15	MalwareEvent	Community < Not published

FIGURE 13. SAMPLE LIST OF EVENTS IN MISP

To create an event in MISP, operators will need, at a minimum, the name and description. Operators should add as much information as possible from the malicious activity report that may be used to add new threat actors, malware, or campaigns. Figure 14 displays the GUI for adding an Event.

The event created will be visible to the organisations having an account on this platform, but not synchronised to other MISP instances until it is published.

List Events
Add Event
Import from...
REST client
List Attributes
Search Attributes
View Proposals
Events with proposals
View delegation requests
Export
Automation

Add Event

Date: 2021-03-23

Distribution: This community only

Threat Level: High

Analysis: Initial

Event Info: Quick Event Description or Tracking Info

Extends Event: Event UUID or ID. Leave blank if not applicable.

Submit

FIGURE 14. ADDING AN EVENT IN MISP

It's not hard to add additional attributes to an event such as category, type, distribution, and value (see Figure 14).

The screenshot shows the 'Add Attribute' form in the MISP interface. The left sidebar contains navigation links: List Events, Add Event, Import from..., REST client, List Attributes, Search Attributes, View Proposals, Events with proposals, View delegation requests, Export, and Automation. The main form area has the following fields:

- Category**: A dropdown menu with '(choose one)' selected.
- Type**: A dropdown menu with '(choose category first)' selected.
- Distribution**: A dropdown menu with 'Inherit event' selected.
- Value**: A large text area containing a red box with the text 'Add IOCs Here'.
- Contextual Comment**: A text input field.
- Checkboxes**: Three checkboxes labeled 'for Intrusion Detection System', 'Batch Import', and 'Disable Correlation'.
- First seen date** and **Last seen date**: Date pickers.
- First seen time** and **Last seen time**: Time pickers with the format 'HH:MM:SS.ssssss+TTTT'.
- Expected format**: HH:MM:SS.ssssss+TTTT (shown for both time pickers).
- Submit**: A blue button at the bottom.

FIGURE 15. ADDING ATTRIBUTE DATA TO AN EVENT IN MISP

See Figure 16 for the Attribute Category dropdown menu, which presents a list of 14 categories.

- | | |
|--------------------------|---------------------|
| 1. Internal Reference | 8. Network Activity |
| 2. Targeting Data | 9. Payload Type |
| 3. Antivirus Detection | 10. Attribution |
| 4. Payload Delivery | 11. Support Tool |
| 5. Artifacts Dropped | 12. Social Network |
| 6. Payload Installation | 13. Person |
| 7. Persistence Mechanism | 14. Other |

FIGURE 16. CATEGORIES IN MISP

Operators may add attributes to the event (see Figure 17), such as IOCs and additional information:

- **Category:** Where the information such as external source was found
- **Type:** Kind of data to be added
- **Distribution:** All communities, or if chosen manually, a distribution will be restrictive
- **Value:** Adding a batch of IOCs
- **Contextual Comment:** Additional information that is complementary to the event
- **For IDS:** Check mark this box to set the IDS flag if applicable
- **Batch Import:** Check mark this box to add a batch of IOCs of the same category and type

Figure 17 shows the event, associated attributes, and tags that will enable operators to search in MISP for specific adversary threat information that can now be added to the Cyber Hunt Plan to support the Malicious Activity Model.



FIGURE 17. EXAMPLE OF HOW TO ADD TAGS TO AN EVENT IN MISP

Searching in MISP

In this section, operators will be shown how to search MISP for cyber threat data associated with the adversary. The Cyber Hunt Plan from Step 1: Creating a Malicious Activity Model (page 16), will be used to search in MISP for cyber threat data associated with APT29 which is associated to the T1053 Scheduled Task (or schtask) technique.

TABLE 4. CYBER HUNT PLAN

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.

Start by logging into the MISP instance. Click List Events. Find the search function on the right-hand side of the MISP screen, below the login name. Type APT29 and click Filter. Five records of APT29 are identified in the instance. Review these records to learn about associated attributes of APT29 to support the malicious activity hypothesis in Table 4.



FIGURE 18. SEARCHING FOR APT29 IN MISP

The next example will search for T1053. Click List Events, and in the search bar, type in T1053. A total of nine records appears in the instance. Operators will want to review each record cluster for APT29 (see Figure 19).

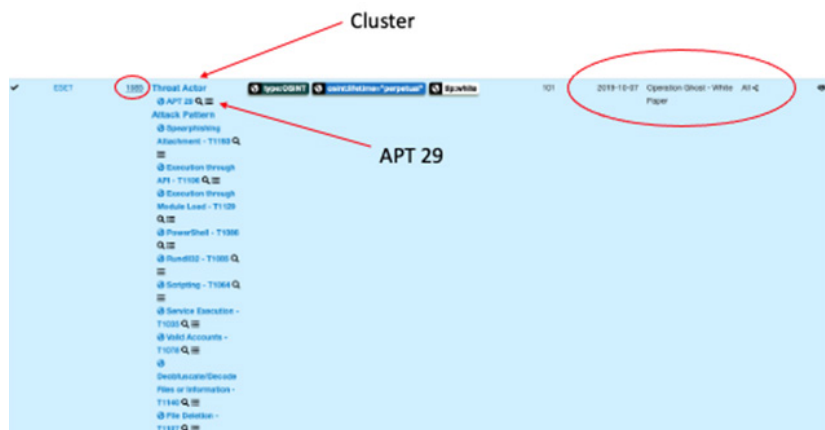


FIGURE 19. SEARCHING FOR SCHEDULED TASK TECHNIQUE T1053 AND APT29 IN MISP

The red circle around number 1085 identifies the Event ID. The red circle on the right is the associated information (which is a paper) for this event. Select Event ID 1085 to expand the correlated information associated with the entire event.

Figure 20 is all the information associated with Event ID 1085.

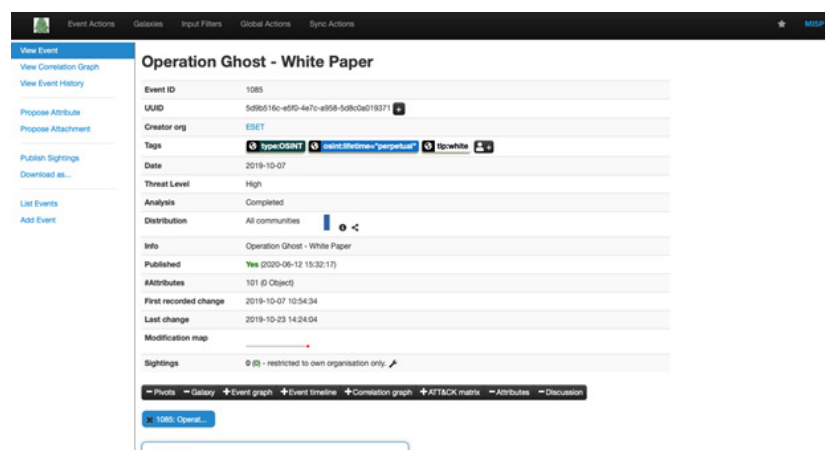


FIGURE 20. SEARCHING FOR T1053 AND APT29 IN MISP

Continue scrolling down toward the bottom of the screen past the galaxies. The screen will display all the correlated events data for Operation Ghost. See Figure 21 as an example.

Date	Org	Category	Type	Value	Tags	Galaxies	Comment	Correlate	Related Events	Feed hits	Distribution
2019-10-23	External analysis	link		https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf			Research White Paper				Inherit
2019-10-16	Network activity	url		http://www.evernote.com/shard/s675/sh/5686fa-889b-49db-8c0b-428f0c04d8/n/7b6dc820f1730147235a95d31a80f			Public webpage used by PolyglotDuke				Inherit
2019-10-07	Network activity	domain		bandabonga.fr			LinDuke C&C				Inherit
2019-10-07	Network activity	domain		westmediasgroup.net			FatDuke C&C				Inherit
2019-10-07	Network activity	domain		shagyniga.com			FatDuke C&C				Inherit
2019-10-07	Network activity	domain		ministernetwork.org			FatDuke C&C				Inherit
2019-10-07	Network activity	domain		tarfedtech.org			FatDuke C&C				Inherit
2019-10-07	Network activity	domain		bussaylawoffice.com			FatDuke C&C				Inherit
2019-10-07	Network activity	domain		spawappliance.com			MinDuke C&C				Inherit

FIGURE 21. CORRELATED DATA ASSOCIATED WITH OPERATION GHOST

Click the first highlighted value to retrieve the Operation Ghost White Paper.pdf. A screen shot of the front cover of the paper is provided in Figure 22 below.



FIGURE 22. COVER OF OPERATION GHOST WHITE PAPER

THE APT GROUPS COMPROMISED GOVERNMENT SYSTEMS, INCLUDING THOSE OF THREE EUROPEAN MINISTRIES OF FOREIGN AFFAIRS.

This white paper presents well-documented threat information regarding APT29, both of which were running successful espionage campaigns at the time. The APT groups compromised government systems, including those of three European Ministries of Foreign Affairs. They also conducted phishing attempts against the Labour Party and the Armed Forces of Norway.

Operators will want to read the white paper, specifically the sections on Tactics and Tools and Section 8: MITRE ATT&CK Techniques. Section 8 reviews in detail the tactics and techniques used by APT29.

As an example, opening the Operation Ghost White paper to page 39, Section 8: MITRE ATT&CK Techniques, will list the technique T1053 Scheduled Task (in the Tactics section, under Persistence). The description states: “The Dukes (also known as APT29 or Cozy Bear) use Scheduled Task to launch malware at startup” (Faou, Tartare, & Dupuy, 2019, p. 39) (see Figure 23). Therefore, T1053 will be linked to the technique T1060 Registry Run Keys/Startup Folder. Both techniques are being used to run malware at startup on the computer.

So, let’s summarize. Operators have searched in MISP for both APT29 and the technique T1053: Scheduled Task. A record was identified, Event ID 1085, the cluster associated with APT29. The red circle on the right of Figure 19 was the information on a paper named, “Operation Ghost—White Paper.” Figure 21 shows the expanded event, with correlated information that needs to be reviewed to support the Cyber Hunt Plan.

Note: Not all information papers that are linked to MISP will have correlated TTPs to the MITRE ATT&CK Framework. However, if the COI has included the associated events and attributes, within the MISP application itself, MISP will highlight those TTPs. An example will be shared at the end of this section. Figure 23 shows details of the search that will be relevant to the Cyber Hunt Plan.

8. MITRE ATT&CK TECHNIQUES

Tactic	ID	Name	Description
Initial Access	T1093	Spearphishing Attachment	The Dukes likely used spearphishing emails to compromise the target.
	T1078	Valid Accounts	Operators use account credentials previously stolen to come back on the victim's network.
Execution	T1066	Execution through API	They use CreateProcess or LoadLibrary Windows APIs to execute binaries.
	T1029	Execution through Module Load	Some of their malware load DLL using LoadLibrary Windows API.
	T1086	PowerShell	FatDuke can execute PowerShell scripts.
	T1085	Rundll32	The FatDuke loader uses rundll32 to execute the main DLL.
	T1064	Scripting	FatDuke can execute PowerShell scripts.
	T1035	Service Execution	The Dukes use PsExec to execute binaries on remote hosts.
	T1060	Registry Run Keys / Startup Folder	The Dukes use the CurrentVersion\Run registry key to establish persistence on compromised computers.
Persistence	T1053	Scheduled Task	The Dukes use Scheduled Task to launch malware at startup.
	T1078	Valid Accounts	The Dukes use account credentials previously stolen to come back on the victim's network.
	T1084	Windows Management Instrumentation Event Subscription	The Dukes used WMI to establish persistence for RegDuke.
	T1040	Deobfuscate/Decode Files or Information	The droppers for PolyglotDuke and LiteDuke embed encrypted payloads.
	T1077	File Deletion	The Dukes malware can delete files and directories.

FIGURE 23. SCREEN CAPTURE OF MITRE ATT&CK TECHNIQUES, P. 39

Next, in the Operation Ghost White paper, on page 24, [Section 4.5](#) Installation and Persistence, will show that the adversary modified the key registry and created a new value named Canon Gear under C:\Program Files....(see Figure 24) (Faou, Tartare, & Dupuy, 2019).

Installation and persistence

During our investigation, we were not able to find a dropper for FatDuke. We believe the operators simply install the backdoor and establish persistence using the standard commands of an earlier stage backdoor.

We also noted that FatDuke generally replaced the second-stage binary, reusing the persistence mechanism already in place.

The persistence we have seen is very standard. They use the registry key `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` and create a new value named `Canon Gear` and value `C:\Program Files\Canon\Network ScanGear\Canocpc.exe`. This launches the backdoor each time a user logs in.

FIGURE 24. SCREEN CAPTURE FROM OPERATION GHOST WHITE PAPER, P. 24

Based on this information, if operators run the autoruns tool, it can identify if this registry key and value are present within the network. Operators will need to run autoruns on each host separately. It's good practice to use autoruns periodically to maintain awareness within the network and ensure that systems stay within the appropriate baselines (Sophos, 2021).

Mapping MISP Event Attributes to ATT&CK Framework

Finally, MISP has the ability to correlate events and their attributes to the MITRE ATT&CK for Enterprise Framework automatically. MITRE highly encourages operators to understand how to use this framework in the discovery process of an adversary TTP. Figure 25 references Operation Ghost, which is the APT29 example used this section. See the red circle around “ATTACK Matrix” link.



FIGURE 25. OPERATION GHOST WHITE PAPER CORRELATED EVENTS

When selecting the ATTACK Matrix, the correlated events and attributes will automatically highlight the TTPs used to execute Operation Ghost with the MITRE ATT&CK for Enterprise Matrix as shown in Figure 26.

MITRE ATT&CK for Enterprise Framework										1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100										
Initial access (T1)										Execution (T1)										Privilege escalation (T1)										Defense evasion (T1)										Credential access (T1)										Discovery (T1)										Lateral movement (T1)										Collection (T1)										Command and control (T1)										Impact (T1)																													
Phishing										Execution through API										Registry Run Keys / Startup Folder										Scheduled Task										Connection Proxy										Account Manipulation										File and Directory Discovery										Windows Admin Shares										Data from Local System										Connection Proxy										Exfiltration Over Command and Control Channel										Account Access Removal									
Valid Accounts										Execution through Remote Link										Valid Accounts										Valid Accounts										Desktop/Device Files or Information										Batch History										Network Share Discovery										Application Scripting										Data from Network Based Drive										Data Obfuscation										Automated Exfiltration										Data Destruction									
Directory Compromise										Powercat										Valid Accounts										Access Token Manipulation										File Collection										Stole Force										Process Discovery										Application Access Token										Data from Remote Host										Fallback Channels										Data Compressed										Data Encrypted for Exfiltration									
Exploit Public-Facing Application										Run/Hide										Windows Management Instrumentation Event Subscription										Accessibility Features										Memory Registry										Cloud Instance Metadata API										System Network Connections Discovery										Application Deployment Software										Audio Capture										Standard Application Layer Protocol										Data Encrypted										Data Removal									
External Remote Services										Scheduled Task										Both profile and service										AppCert DLLs										Unauthorized Files or Information										Credential Dumping										Account Discovery										Component Object Model and Distributed COM										Automated Collection										Web Service										Data Transfer Size Limits										Disk Content Wipe									
Hardware Additions										Scripting										Accessibility Features										Applet DLLs										Run/Hide										Credentials from Web Browsers										Application Window Discovery										Exploitation of Remote Service										Clipboard Data										Commonly Used Port										Exfiltration Over Alternative Protocol										Disk Structure Wipe									
Replication Through Removable Media										Service Execution										Account Manipulation										Application Shimming										Scripting										Credentials in Files										Browser Bookmark Discovery										Internal Spearphishing										Data Staged										Communication Through Removable Media										Exfiltration Over Other Network Medium										Endpoint Denial of Service									
Speepphishing Link										AppletScript										AppCert DLLs										Bypass User Account Control										Software Packing										Credentials in Registry										Cloud Service Dashboard										Login Scripts										Data from Cloud Storage Object										Custom Command and Control Protocol										Exfiltration Over Custom Protocol										Firmware Corruption									
Speepphishing via Service										QAMTP										Applet DLLs										DLL Search Order Hijacking										Valid Accounts										Exploitation for Credential Access										Cloud Service Discovery										Pass the Hash										Data from Information Repositories										Custom Cryptographic Protocol										Scheduled Transfer										Initial System Recovery									
Supply Chain Compromise										Command Line Interface										Application Shimming										Dylib Hijacking										Web Service										Forced Authentication										Domain Trust Discovery										Pass the Ticket										Email Collection										Data Encoding										Transfer Data to Cloud Account										Network Denial of Service									
Trusted Relationship										Compiled HTML File										Authentication with Prompt										Elevated Execution with Prompt										Access Token Manipulation										Hooking										Network Service Scanning										Remote Desktop Protocol										Input Capture										Domain Fronting										Resource Hijacking																			

Download MITRE ATT&CK for Enterprise Framework

This is an initial attempt. Please contact your system administrator for more information. MITRE ATT&CK for Enterprise Framework

FIGURE 26. CORRELATED DATA TO THE MITRE ATT&CK FOR ENTERPRISE FRAMEWORK

Note: The current ATT&CK Matrix MISP is using in their application must be updated to the newest version of ATT&CK model. Visit <https://attack.mitre.org> for the latest version.

This summarizes how to use the CIRCL MISP application. Before moving to Step 2, operators should review the MITRE ATT&CK Framework at <https://attack.mitre.org> for the knowledge base of adversary tactics and techniques derived from real-world observation of adversary techniques. Figure 27 shows a screen capture of the latest version.

[illegible]

FIGURE 27. MITRE ATT&CK MATRIX

STEP 2: DEVELOP HYPOTHESES AND ABSTRACT ANALYTICS

Defining a Hypothesis and Resources to Inform Hypothesis and Analytic

A *hypothesis* for a cyber hunt is the educated belief that an adversary will behave in a certain way, and an *analytic* is the method used to detect the adversary behavior identified in the hypothesis. For example, if the hypothesis is that an adversary will schedule malware installations using *schtasks* (Windows command for scheduling commands and programs) through the malicious activity model, an analytic needs to be developed to detect that *schtasks* is being run.

MITRE has identified three resources that could help inform the hypothesis and analytic: adversary emulation plans, the MITRE CAR, and MITRE ATT&CK. Several references are listed below to aid in developing a hypothesis and analytics:

- **Adversary Emulation APT3:** <https://attack.mitre.org/resources/adversary-emulation-plans/>
- **Adversary Emulation APT29:** https://github.com/mitre-attack/attack-arsenal/blob/master/adversary_emulation/APT29/Emulation_Plan/APT29_EmuPlan.pdf
- **MITRE ATT&CK:** <https://attack.mitre.org/>
- **MITRE CAR:** <https://car.mitre.org/>

Additional resources for reference:

- **Azure Sentinel Hunting Queries:** <https://github.com/Azure/Azure-Sentinel/tree/master/Hunting%20Queries>
- **Elastic Detection Rules:** <https://github.com/elastic/detection-rules>
- **Event Query Language Analytics Library:** <https://eqllib.readthedocs.io/en/latest/analytics.html>
- **Falcon Force Friday Github:** <https://github.com/FalconForceTeam/FalconFriday>
- **MAGMA Use Case Framework:** <https://www.betaalvereniging.nl/en/safety/magma/>
- **Palantir Alerting and Detections Strategies Framework:** <https://github.com/palantir/alerting-detection-strategy-framework>
- **Security Operations Center (SOC) Prime MITRE ATT&CK Map:** <https://attack.socprime.com/#/>
- **Threat Hunter Playbook:** <https://threathunterplaybook.com/introduction.html>
- **Uncoder.io:** <https://uncoder.io/>
- **SigmaHQ:** <https://github.com/SigmaHQ/sigma>

STEP 2 COVERS

- Defining a hypothesis and provide resources to help inform the hypothesis and analytic
- Building a hypothesis and analytic using MITRE's CAR knowledgebase
- Building a hypothesis and analytic using MITRE's ATT&CK
- Starting the Cyber Hunt Plan based on the hypothesis, CAR, and ATT&CK

This technical manual will focus on *APT29 Adversary Emulation Plan* (plan can be found linked in Appendix B) and *ATT&CK*. Additional emulation plans are either published or planned to be published in the future and can be leveraged as well.

The *APT29 Adversary Emulation Plan* is the second in a series of emulation plans that document known APT behaviors through tactics, techniques, and procedures (TTPs) that have been publicly reported. The plan typically uses the ATT&CK Framework to characterize those behaviors, so it's important to be aware of the ATT&CK Framework when reading the emulation plans. MITRE has also included APT28 and APT29 ATT&CK Techniques in Appendix A and Appendix B respectively. Appendix C provides a comparison on the techniques used by APT28 and APT29 so that cyber operators become familiar with these adversary TTPs.

ATT&CK is a knowledge base of known TTPs that are based on real-world observations (see Figure 27 on page 38). These TTPs are used to characterize adversary behaviors and to build threat models and methodologies like Adversary Emulation Plans or CAR.

CAR is a knowledge base of analytics developed by MITRE and a community of users; these analytics are designed to be tool agnostic so that the analytic can be applied to any tool.

Step 2 will cover writing both a hypothesis and an analytic. CAR can provide both of these items for a cyber hunt, but it doesn't necessarily need to be the only source used for this step. It is also important to note that if a hypothesis is used from CAR, the hypothesis should be validated with the malicious activity model developed in Step 1. Given that, an example is provided using CAR and then another example using only ATT&CK TTPs.

For the following examples, ATT&CK and the *APT29 Adversary Emulation Plan* will be used as the sources for the malicious activity model.

A HYPOTHESIS FOR A CYBER HUNT IS THE EDUCATED BELIEF THAT AN ADVERSARY WILL BEHAVE IN A CERTAIN WAY, AND AN ANALYTIC IS THE METHOD USED TO DETECT THE ADVERSARY BEHAVIOR IDENTIFIED IN THE HYPOTHESIS.

Building the Hypothesis and Analytic via MITRE's CAR

Per the ATT&CK knowledge base, it is known that APT29 has a record of using scheduled tasks (T1053); two FireEye researchers documented APT29's use of scheduled tasks as two of their seven unique persistence mechanisms (FireEye, 2020). This known behavior can be searched for in the CAR knowledge base for an analytic that will correspond to this behavior.

Reviewing the analytics that in CAR (<https://car.mitre.org/analytics/>), two tables are shown that list all of the analytics in CAR. Figure 28 displays a plain list of the analytics.

MITRE Cyber Analytics Repository					
Analytics Data Model Sensors Coverage Comparison					
Analytics					
Analytic List (sortable)					
ID ▾	Name	Submission Date	ATT&CK Techniques	Implementations	Applicable Platforms
CAR-2021-02-002	Get System Elevation	January 15 2021	<ul style="list-style-type: none"> Abuse Elevation Control Mechanism 	Pseudocode, Splunk	Windows
CAR-2021-02-001	Webshell-Indicative Process Tree	November 29 2020	<ul style="list-style-type: none"> Server Software Component 	Pseudocode, Splunk	Windows
CAR-2021-01-009	Detecting Shadow Copy Deletion via Vssadmin.exe	December 11 2020	<ul style="list-style-type: none"> Inhibit System Recovery 	Splunk	Windows
CAR-2021-01-008	Disable UAC	December 11 2020	<ul style="list-style-type: none"> Abuse Elevation Control Mechanism 	Pseudocode, Splunk	Windows

FIGURE 28. CAR ANALYTICS LIST

This list includes all the analytics (with more detailed information by selecting the CAR ID) in a table format that can be sorted by each column (e.g., by Submission Date, Implementation). Figure 29 displays the second table listed on the analytics site, which organizes the same analytics by the ATT&CK Techniques.

Analytic List (by technique/sub-technique coverage)		
ATT&CK Technique	ATT&CK Sub-technique(s)	CAR Analytic(s)
Create or Modify System Process	Windows Service	<ul style="list-style-type: none"> CAR-2013-01-002: Autorun Differences CAR-2013-04-002: Quick execution of a series of suspicious commands CAR-2013-09-005: Service Outlier Executables CAR-2014-02-001: Service Binary Modifications CAR-2014-03-005: Remotely Launched Executables via Services CAR-2014-05-002: Services launching Cmd
Scheduled Task/Job	(N/A - see below)	(N/A - see below)
...	Scheduled Task	<ul style="list-style-type: none"> CAR-2013-01-002: Autorun Differences CAR-2013-04-002: Quick execution of a series of suspicious commands CAR-2013-08-001: Execution with schtasks CAR-2015-04-002: Remotely Scheduled Tasks via Schtasks CAR-2020-09-001: Scheduled Task - FileAccess
...	At (Windows)	<ul style="list-style-type: none"> CAR-2013-04-002: Quick execution of a series of suspicious commands CAR-2013-05-004: Execution with AT CAR-2015-04-001: Remotely Scheduled Tasks via AT

FIGURE 29. CAR ANALYTICS LIST, ORGANIZED BY ATT&CK TECHNIQUE

Let's search CAR for an analytic that corresponds to the identified adversary behavior (T1053—Scheduled Task). Searching for scheduled task results in five CAR analytics listed in the table (see the highlights in Figure 30).

Analytic List (by technique/sub-technique coverage)		
ATT&CK Technique	ATT&CK Sub-technique(s)	CAR Analytic(s)
Create or Modify System Process	Windows Service	<ul style="list-style-type: none">CAR-2013-01-002: Autorun DifferencesCAR-2013-04-002: Quick execution of a series of suspicious commandsCAR-2013-09-005: Service Outlier ExecutablesCAR-2014-02-001: Service Binary ModificationsCAR-2014-03-005: Remotely Launched Executables via ServicesCAR-2014-05-002: Services launching Cmd
Scheduled Task/Job	(N/A - see below)	(N/A - see below)
...	Scheduled Task	<ul style="list-style-type: none">CAR-2013-01-002: Autorun DifferencesCAR-2013-04-002: Quick execution of a series of suspicious commandsCAR-2013-08-001: Execution with schtasksCAR-2015-04-002: Remotely Scheduled Tasks via SchtasksCAR-2020-09-001: Scheduled Task - FileAccess
...	At (Windows)	<ul style="list-style-type: none">CAR-2013-04-002: Quick execution of a series of suspicious commandsCAR-2013-05-004: Execution with ATCAR-2015-04-001: Remotely Scheduled Tasks via AT

FIGURE 30. ATT&CK: SCHEDULED TASK ANALYTICS

Each analytic can be viewed by navigating to each of the CAR IDs located in the far-right column. Each analytic includes a hypothesis, summary data, ATT&CK detection data, CAR Data Model references, and implementations.

MITRE Cyber Analytics Repository
Analytics
Data Model
Sensors
Coverage Comparison

CAR-2013-08-001: Execution with schtasks

The Windows built-in tool `schtasks.exe` provides the creation, modification, and running of [scheduled tasks](#) on a local or remote computer. It is provided as a more flexible alternative to `at.exe`, described in [CAR-2013-05-004](#). Although used by adversaries, the tool is also legitimately used by administrators, scripts, and software configurations. The scheduled tasks tool can be used to gain [Persistence](#) and can be used in combination with a [Lateral Movement](#) technique to remotely gain [execution](#). Additionally, the command has parameters to specify the user and password responsible for creating the task, as well as the user and password combination that the task will run as. The `/s` flag specifies the remote system on which the task should be scheduled, usually indicating [Lateral Movement](#).

Submission Date: 2013/08/07
Information Domain: Host
Data Subtypes: Process
Analytic Type: TTP
Applicable Platforms: Windows
Contributors: MITRE

ATT&CK Detection

Technique	Subtechnique(s)	Tactic(s)	Level of Coverage
Scheduled Task/Job	Scheduled Task	Persistence	Moderate

Data Model References

Object	Action	Field
process	create	exe
process	create	command_line

Implementations

Pseudocode

Look for instances of `schtasks.exe` running as processes. The `command_line` field is necessary to disambiguate between types of `schtasks` commands. These include the flags `/create`, `/run`, `/query`, `/delete`, `/change`, and `/end`.

```

process = search Process:Create
schtasks = filter process where (exe == "schtasks.exe")
output schtasks

```

FIGURE 31. CAR ANALYTIC—SCHEDULED TASK

The CAR analytic (see Figure 31) is a great example for Step 2 since a hypothesis and an analytic has been identified that can be used to inform the hands-on hunt later in the TTP-Based Hunt methodology. The hypothesis is identified in the first paragraph, and the analytic is identified in the implementation pseudocode located at the bottom. Operators can take this pseudocode and apply it to the tools that will be used in later steps

Building the Hypothesis and Analytic via ATT&CK

There may be times when malicious behavior (from the malicious activity model) does not have a corresponding analytic in CAR, and a new hypothesis and analytic must be created. To make this easier, operators can leverage the ATT&CK Framework to characterize the malicious behavior.

To build a hypothesis and analytic, the APT29 Adversary Emulation Plan will be used as the baseline. In the second step of the first scenario of the Adversary Emulation Plan, researchers have identified that APT29 uses

BECAUSE THE
ADVERSARY EMULATION
PLANS USE THE ATT&CK
FRAMEWORK TO
DESCRIBE ADVERSARY
BEHAVIORS, THE
OPERATORS CAN EASILY
DEVELOP A HYPOTHESIS
AND ANALYTIC
USING THE ATT&CK
FRAMEWORK AS WELL.

pupy to exfiltrate files and documents that they have previously identified (see Figure 32). *Pupy* is an open-source, cross-platform (Windows, Linux, OSX, and Android) remote administration and post-exploitation tool used in several TTP's identified in the MITRE ATT&CK framework (MITRE, 2018).

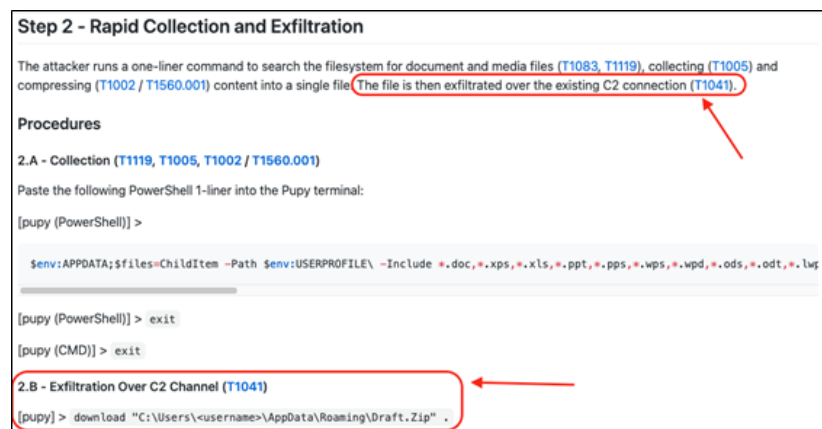


FIGURE 32. APT29 ADVERSARY EMULATION PLAN—ATT&CK TECHNIQUE

Because the Adversary Emulation Plans use the ATT&CK Framework to describe adversary behaviors, the operators can easily develop a hypothesis and analytic using the ATT&CK Framework as well. Figure 32 shows the summary section of Step 2 in the Adversary Emulation Plan. The report describes the adversary behavior with ATT&CK identifiers (e.g., T1119, T1005). Operators can use these references to learn more about these behaviors and write better hypotheses and analytics. Operators can also define suspected future adversary behavior (hypotheses) and build abstract detection methods to discover this behavior in future hunts.

Here is the hypothesis based on this malicious behavior: It is *suspected the adversary has established a command-and-control channel and that they will attempt to download files using pupy, or a similar remote administration tool, through this channel. This practice corresponds to the ATT&CK Technique: T1041, Exfiltration Over Command and Control (C2) Channel. An abstract analytic for this behavior would be to detect anomalous File Transfer Protocol (FTP) activity at unusual times, large FTP traffic, or unknown users.*

In the context of this example, to make this hypothesis more comprehensive operators should consider the following:

- Is FTP authorized on the network?
- At what volume is FTP nominal during business hours and is FTP expected outside of business hours?
- Are certain users or certain systems authorized to use FTP?
- How can adversaries evade this hypothesis using other network protocols or manipulating file sizes?

It's important to keep the analytic at an abstract level at this point in the methodology to account for any changes later in the methodology.

Building the Cyber Hunt Plan

Now that operators have either (a) used an existing analytic repository like CAR to identify an analytic and hypothesis or (b) created a hypothesis and an analytic from scratch using ATT&CK characterizations, they can extract required data to instantiate these analytics and develop the Cyber Hunt Plan. At this point in the plan, operators have built a malicious activity model and created a hypothesis with an abstract analytic (see Table 5).

TABLE 5. CYBER HUNT PLAN UPDATE—HYPOTHESES & ABSTRACT ANALYTICS

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.
Hypotheses and Abstract Analytics	It is suspected that the adversary has used scheduled tasks to establish persistence. CAR analytic <i>CAR-2013-08-001</i> can help hunt for this suspicion.

At this point, operators are still on the left side of the TTP-Based Hunting methodology (Reference Figure 3), or *Characterization*, and may need to iterate back to the malicious activity model if the hypothesis and analytic need further refinement. This is an iterative process.

The next section will discuss how data requirements will be determined given the identified analytics and hypotheses from the hunt plan. At this point, the technical details of data gathering will need to be considered (e.g., endpoint data vs. network data).

STEP 3: DETERMINE DATA REQUIREMENTS

STEP 3 COVERS

- Providing an overview of data types that can be leveraged for the malicious activity model and/or cyber hunt plan (Endpoint, Network, and Security)
- Leveraging ATT&CK data sources
- Determining Sysmon data requirements for T1053
- Taking advantage of additional resources

An Overview of Data Types

Cyber threat hunters rely on the ability for their IT organization to ingest and index a wide variety of data types from multiple data sources at near real-time speed. Most important, is the organization's ability to collect quality data to enhance their threat hunting capabilities. Below is a list of data types to be considered:

- **Endpoint data** comes from end user devices within the network such as desktop computers, laptops, and mobile phones. Some organizations include their data center hardware in this list. Examples of key data from these endpoints:
 - **Registry data** is related to registry objects, including key and value metadata on Windows-based endpoints.
 - **File data** includes dates when files on an endpoint were created or modified, as well as their type, size, and location within the disk.
 - **Process execution metadata** contains information on the different processes running on the endpoints, as well as command-line commands, arguments, process file names and IDs.
 - File prevalence refers to the number of users (devices) that have seen the file and how long the users have seen the file in the environment.
- **Network data** comes from devices such as firewalls, switches, routers, proxy servers, and domain name server (DNS). Examples of key data from these devices:
 - **DNS logs** host data related to DNS resolution, which may contain domain-to-IP address mappings and identification of internal clients are making resolution requests.
 - **Network data** refers to connection/session information between hosts on the network (e.g., source and destination IP address, connection duration times from start to end, netflow).
 - **Proxy log data** refers to Hypertext Transfer Protocol (HTTP) data containing information on outgoing web requests on internet resources that are being accessed within the internal network.
 - **Monitoring log data** comes from monitoring tools that will collect application metadata such as HTTP, DNS, and Simple Mail Transfer Protocol (SMTP) as well as connection-based data flow logged data.

- **Firewall logs** document how the firewall manages traffic types. Logs offer insight into source and destination IP addresses, protocols, and port numbers. This data can be one of the most important types of data that is collected.
- **Switch and router logs** contain basic information about network traffic. Logs contain data and timestamp, IP addresses, and Transmission Control Protocol (TCP) ports, action the device took based on Access Control List, size of packet, etc.
- **Security data** come from internal security devices, such as Security Information and Event Management (SIEM) system, IPS, and IDS solutions that may have been deployed on the network. Examples of key data from these devices:
 - **Alerts** are notifications from IDS, IPS and SIEM solutions indicating that a ruleset was violated, or an incident has occurred.
 - **Threat Intelligence** is Information from threat intelligence feeds such as MISP or other open-source feeds regarding adversary TTPs and their behavior.

It's important to note that traditional Defensive Cyber Operations (DCO) methodologies log all data sources and analyze that data at a later time. Using this approach leads to potential overload of the logging, transport, and ingestion devices through increased processing, bandwidth, and storage requirements and to operator overload without effective data triage. Defining specific data requirements and comparing those requirements to the logging capabilities of available sensors will reduce this overload and provide a usable system. To this end, it is important to document the available data sources that are being collected and understand what that data is and if that tool is collecting the best data possible for the engagement. For example, if Sysmon is being collected, the operator must consider if it is generating the proper event IDs to detect a specific TTP as it logs process creation, Windows endpoint data. This can be a powerful technique to generate requirements like the ones below for more mature hunt tools on the network:

- Understand what operators want to monitor
- Identify tools that can generate the required data
- Understand the capability of the tool

THE ATT&CK KNOWLEDGE BASE INCLUDES METADATA TO HELP DETECT AND IDENTIFY ADVERSARY TTPS.

- Identify events the tools are generating
- Verify the tools are generating the appropriate data for the current use case
- Tune the tool/sensor if needed to generate the required data
- Verify data

While Sysmon was used in this example, this process can be applied to other data sources that may be collected whether the telemetry is endpoint, network, or security data.

Now that different categories of network data that can be collected have been identified, leveraging MITRE ATT&CK and other data sources will aid in the continued development of the Cyber Hunt Plan hypothesis in Table 6.

Leveraging ATT&CK Data Sources

The ATT&CK knowledge base includes metadata to help detect and identify adversary TTPs. This section will show how operators can leverage ATT&CK Data Sources to better understand what types of data are relevant for an ATT&CK TTP. This example will focus primarily on ATT&CK T1053.005.

In ATT&CK version 9 (released in April 2021), data sources have expanded to include more detailed information in YAML Ain't Markup Language (YAML) files, a human-readable data-serialization language. These files are tool agnostic so that they are easily transferable to the SIEM or other tools for a given hunt. To view the ATT&CK Data Sources, navigate to the ATT&CK TTP page (T1053.005, per Figure 33) and view the metadata block, which is usually in the top right-hand corner of the TTP page.

Home > Techniques > Enterprise > Scheduled Task/Job > Scheduled Task

Scheduled Task/Job: Scheduled Task

Other sub-techniques of Scheduled Task/Job (7)

Adversaries may abuse the Windows Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are multiple ways to access the Task Scheduler in Windows. The `schtasks` can be run directly on the command line, or the Task Scheduler can be opened through the GUI within the Administrator Tools section of the Control Panel. In some cases, adversaries have used a .NET wrapper for the Windows Task Scheduler, and alternatively, adversaries have used the Windows `netapi32` library to create a scheduled task.

The deprecated `at` utility could also be abused by adversaries (ex: `at (Windows)`), though `at.exe` can not access tasks created with `schtasks` or the Control Panel.

An adversary may use Windows Task Scheduler to execute programs at system startup or on a scheduled basis for persistence. The Windows Task Scheduler can also be abused to conduct remote Execution as part of Lateral Movement and also run a process under the context of a specified account (such as SYSTEM).

ID: T1053.005
Sub-technique of: T1053
 ① **Tactics:** Execution, Persistence, Privilege Escalation
 ① **Platforms:** Windows
 ① **Permissions Required:** Administrator
 ① **Data Sources:** [Command:](#) Command Execution, [File:](#) File Modification, [Process:](#) Process Creation, [Scheduled Job:](#) Scheduled Job Creation
 ① **Supports Remote:** Yes
Version: 1.0
Created: 27 November 2019
Last Modified: 30 December 2020
[Version Permalink](#)

Procedure Examples

ID	Name	Description
S0331	Agent Tesla	Agent Tesla has achieved persistence via scheduled tasks. ^[1]
S0504	Anchor	Anchor can create a scheduled task for persistence. ^[2]
S0584	AppleJeu	AppleJeu has created a scheduled SYSTEM task that runs when a user logs in. ^[3]

FIGURE 33. ATT&CK DATA SOURCES

In the Data Sources bullet, a list of data sources that the ATT&CK team determined were necessary for detecting this TTP is provided. For example, T1053.005 contains four data sources:

① **Data Sources:** [Command:](#) Command Execution, [File:](#) File Modification, [Process:](#) Process Creation, [Scheduled Job:](#) Scheduled Job Creation

FIGURE 34. ATT&CK WEBSITE—T1053.005 DATA SOURCES

For each of the data sources, a link to the corresponding YAML file is included. These files can also be found by navigating to the MITRE ATT&CK attack-datasources repository on GitHub: <https://github.com/mitre-attack/attack-datasources>.

These YAML files address the “data components” of the data source (e.g. Scheduled Job Creation data components). In each file, there will be a series of “data components” with the data elements that are important for detecting this TTP. For example, Figure 34 shows the data elements that would help detect a scheduled job creation

```
- name: scheduled job creation
  type: activity
  description: A scheduled job was created, either locally or remotely.
  relationships:
    - source_data_element: user
      relationship: created
      target_data_element: scheduled job
    - source_data_element: process
      relationship: created
      target_data_element: scheduled job
```

FIGURE 35. EXAMPLE DATA COMPONENT FOR T1053.005

Figure 35 shows two separate data element relations that can be used to detect a scheduled job was created, either locally or remotely:

- A user created a scheduled job
- A process created a scheduled job

These data sources are not intended to be enough to write analytics directly but are intended to offer another data point to consider when determining the data requirements. For instance, operators now have a series of data components (in the ATT&CK Data Sources) to consider when determining data sources needed to assist in the hunt of a TTP. This information aims to assist operators developing queries that will come in Step 6: Implement and Test Analytics.

Determining Sysmon Data Requirements for T1053

Earlier in this section, categories of data (e.g., endpoint, network, security data) and specific data sources (e.g., logs, file data) were identified. This information will be applied to Sysmon, a popular, Windows event monitoring service. Sysmon is an open-source, system service and device driver that logs and monitors Windows endpoints; Sysmon events can be viewed in Event Viewer, software that is installed with every Windows computer.

The following instructions and guidance cover Sysmon from a limited perspective, scoping Sysmon's usage to this particular use case. There are plenty of community resources available on the internet that can help operators understand Sysmon and its capabilities. TrustedSec, an information security consulting team, publishes and regularly updates a Sysmon Community Guide that can be extremely helpful: <https://github.com/trustedsec/SysmonCommunityGuide>.

Sysmon can be downloaded on the Microsoft sysinternals website if it's not already installed: <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>. Once the software has been downloaded, it can be installed by executing `sysmon.exe` by double clicking the file. See Figure 36 for Sysmon install package.

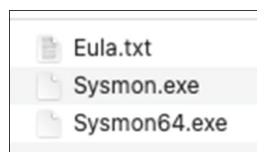


FIGURE 36. SYSMON INSTALL PACKAGE

Alternatively, Sysmon can be installed by using the command line (`cmd.exe`): `sysmon -accepteula -i`. Both installation methods will install the default configurations for Sysmon. This is when data requirements are important for performing the most effective hunt (Russovich & Garnier, Sysmon v13.24, 2021).

To illustrate how the data requirements will be designed, let's look at the Cyber Hunt Plan in Table 6. At this point, three areas in the Cyber Hunt Plan have been identified: (1) the malicious activity model, (2) hypothesis based on that model, and (3) abstract analytic that will help search for that hypothesis. In the pseudocode of the analytic, *CAR-2013-08-00*, operators will need to "look for instances of `schtasks.exe` running as processes," as shown below example in Figure 37.

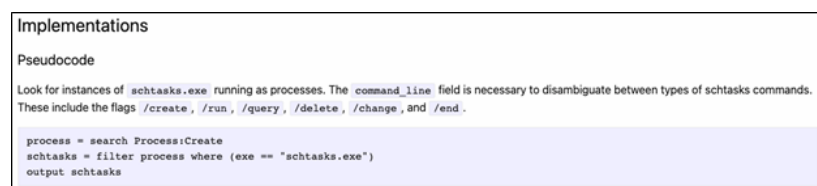


FIGURE 37. CAR-2013-08-00 ANALYTIC IMPLEMENTATION PSEUDOCODE

This means that operators need to determine how Sysmon will look for instances of `schtasks.exe`. The default configuration of Sysmon is rather limited but the Sysmon Community Guide contains excellent instructions on configuring a Sysmon file tailored to a specific use case. In this case, MITRE recommends using the file written and updated by Security Researcher Olaf Hartong. Hartong's configuration file is extremely thorough and well documented, an excerpt of the 2000+ line file is shown in Figure 38. The entire configuration file can be found on GitHub: <https://github.com/olafhartong/sysmon-modular>. Installation instructions for this custom configuration file can be found in this repository as well.

To confirm that this configuration file will instruct Sysmon to search for the correct data requirements (instances of `schtasks.exe`), operators need to review the XML file that would be installed: `sysmonconfig.xml`. Luckily, the configuration file is documented with the correlated ATT&CK techniques, so it's best to search the file based on the ATT&CK technique identifier, T1053, and read the XML configuration. For this example operators should look for T1053 configurations including `schtasks.exe` (Hartong, 2020).

```
<OriginalFileName name="technique=T1053,technique_name=At.exe Periodic Scheduled Task" condition="contains">At.exe</OriginalFileName>
<OriginalFileName name="technique_id=T1016,technique_name=System Network Configuration Discovery" condition="contains any">nbstat.exe;nbinfo.
<OriginalFileName name="technique_id=T1057,technique_name=Process Discovery" condition="is">qwinsta.exe</OriginalFileName>
<OriginalFileName name="technique_id=T1057,technique_name=Process Discovery" condition="is">rwinsta.exe</OriginalFileName>
<OriginalFileName name="technique_id=T1053,technique_name=Scheduled Tasks" condition="contains any">schtasks.exe;sctasks.exe</OriginalFileName>
```

FIGURE 38. SYSMON CONFIGURATION EXCERPT

In the excerpt above, two configuration lines correspond to T1053 in the first and last line of the image. The first configuration is monitoring for `At.exe` where the last line is monitoring for both `schtasks.exe` and `sctasks.exe` confirming that this configuration of Sysmon will correctly search for the data requirements that are identified in the analytic for the Cyber Hunt Plan.

TABLE 6. CYBER HUNT PLAN UPDATE—DETERMINE DATA REQUIREMENTS

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.
Hypotheses and Abstract Analytics	It is suspected that the adversary has used scheduled tasks to establish persistence. CAR analytic <i>CAR-2013-08-001</i> can help hunt for this suspicion.
Determine Data Requirements	Sysmon configuration contains the correct data requirements based on the CAR analytic: instances of <code>schtasks.exe</code> running as processes.

Taking Advantage of Additional Resources

Determining data requirements could be more difficult for further analytics besides the example above. There will be more data sourcing tools beyond Sysmon that need data requirements (e.g. Zeek). The additional resources below are intended to help determine data requirements and navigate Sysmon to understand the data available on Windows endpoints:

- **Atomic Threat Coverage:** <https://github.com/atc-project/atomic-threat-coverage>
- **Malware Archaeology Cheat Sheet:** <https://static1.squarespace.com/static/552092d5e4b0661088167e5c/t/5b8f091c0ebbe8644d3a886c/1536100639356/Windows+ATT%26CK+Logging+Cheat+Sheet+ver+Sept+2018.pdf>
- **SwiftOnSecurity Sysmon Config:** <https://github.com/SwiftOnSecurity/sysmon-config>

STEP 4: FILTERING THE SOURCES OF DATA

STEP 4 COVERS

- Identifying information/data required from the network owner
- Understanding the elements of time, behavior, and cyber terrain
- Using Nmap and Zeek to begin to filter what teams plan to analyze during the hunt for malicious activity

Information Required from the Network Owner

During the previous steps, a malicious activity model and hypothesis has been developed, and the data requirements identified. To assist with filtering, operators should acquire the information shown in Table 7 from the network owner.

TABLE 7. INFORMATION TO ACQUIRE FROM NETWORK OWNER

Resource	Description
Network Maps/Diagrams	Network-focused resources provide a logical layout of the network on which a hunt will be conducted. Network maps can help identify the most effective places in a network to install network TAPs. Network maps outline the flow of the network traffic and will facilitate hunting.
Configuration Baselines for Workstations, Servers, and Network Devices	Configuration baselines are helpful for understanding what a “known good” system looks like. In other words: What are the approved applications on a system? What are the typical processes that should be running? Which services are supposed to be running?
Network Baselines	Network baselines both identify normal traffic activity, authorized ports and protocols, and abnormal traffic.
Vulnerability Assessment Results	A vulnerability assessment provides a list of current vulnerabilities that can be used to identify weaknesses the adversary might take advantage of. Knowing which systems are vulnerable to an exploit can help focus the analysis effort of the hunt.
Logging Configurations	Current logging configurations help identify which data sources will be available for the hunt (e.g., which event logs, Sysmon, and other forms logs).
Group Policy (GPO)	GPOs provide information on what users are allowed to do within a network. While hunting, suspicious users can be compared against the current GPO to identify if their activity is malicious or benign.
Normal Working Hours	Knowing the normal working hours can help to identify suspicious activity outside the of the identified timeframe.
Crown Jewels Analysis	Crown Jewels Analysis is the process of identifying the critical cyber assets necessary for an organization to meet its mission requirements. See MITRE’s Systems Engineering Guide for Crown Jewels Analysis: https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/crown-jewels-analysis

Understanding Time, Behavior, and Cyber Terrain

Collecting data on networks generates a tremendous amount of data. It's not feasible for teams to have analysts look at every event to effectively identify security related threats. Cyber operators use filtering to provide focus to the analysis space using elements of time, behavior, and cyber terrain.

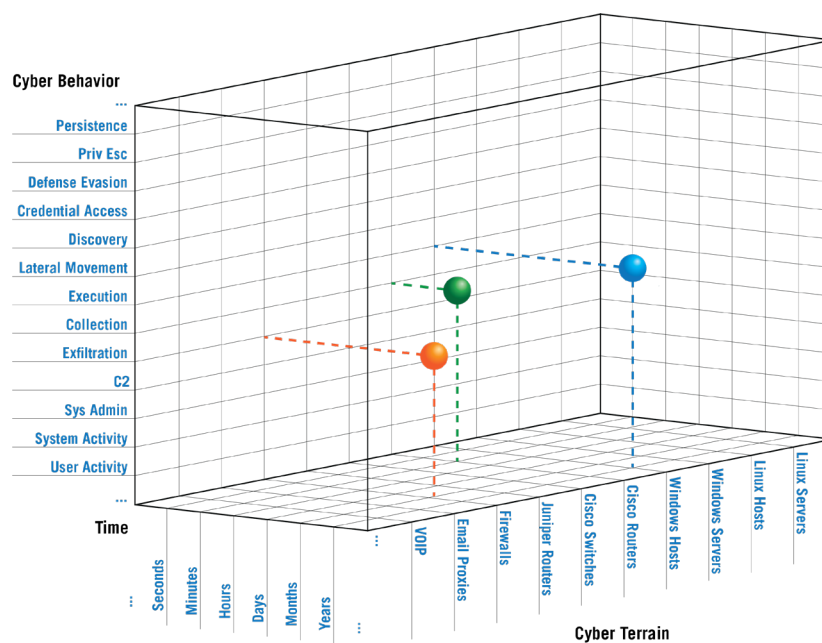


FIGURE 39 ELEMENTS OF TIME, BEHAVIOR, AND CYBER TERRAIN

Operators will use time to filter the activity as shown in Figure 39. This filter can help determine how far back operators should go to conduct a historical search or to narrow the focus of the analysis to include designated timeframes. Time can also be used to determine how much time will be needed to obtain a sampling size required for an analytic to be effective. Some analytics may require 30 days of logs to be effective. It's important to remember the sensing solution will only have data from when the system was deployed. A threat hunt engagement may not be able to review data sets 30 days back until it has had time to collect that data. The challenge is that the new data being collected may not have the artifacts, IOCs, or TTP's that may be present in stored historical data. Stored historical data are the logs of different devices that were stored prior to the deployment of

IT IS IMPORTANT
FOR OPERATORS
TO EVALUATE THE
ANALYTICS DEVELOPED
FOR THE ENGAGEMENT
AND IDENTIFY IF THOSE
ANALYTICS WOULD
DETECT NOMINAL
ACTIVITY THAT COULD
BE ATTRIBUTED TO
APPROVED SYSTEM
ADMINISTRATIVE
ACTIONS.

the sensing solution. If the sensing solution is integrated into a predefined deployment that is collecting the new data, developing a process to ingest the previously stored historical data into the data analytics tool designed may enable operators to potentially detect anomalous behavior. There may be challenges with the way the historical data is normalized by the data analytics tool compared to the newly collected data which can prevent operators to properly filter and search on the stored historical data. All data being ingested need to be properly tested to ensure all the data is able to be properly searched. Being aware of current retention policies on different data types that are being stored for specific time periods can provide insight to different historical data types, which may provide value to a threat hunt engagement.

Behavior can help differentiate normal behaviors (e.g., actions performed by system administrators) from malicious activity. It is important for operators to evaluate the analytics developed for the engagement and identify if those analytics would detect nominal activity that could be attributed to approved system administrative actions. For example, to detect a potentially malicious use of PowerShell will require operators to be aware of nominal PowerShell usage on the network to reduce false positives. It is common for adversaries to leverage legitimate tools that are already present in the network to achieve their goals and evade detection by blending in with normal activity. This is referred to as living off the land. Working with the system administrators and local defenders can assist with developing baselines of activity and new attack vectors. More information on living off the land techniques can be found at <https://github.com/LOLBAS-Project/LOLBAS>.

For the example of finding malicious use of scheduled tasks, operators will want to know what scheduled tasks are already being used in the network. The first option is to ask the network owners which scheduled tasks are already being used and on which systems. This knowledge will help create a baseline of known scheduled tasks.

The second option is to use Autoruns for Windows, which is a tool that is part of the Microsoft Sysinternals Suite. Autoruns identifies programs that are configured to run during system bootup or logins and when various, built-in Windows applications are started. After running Autoruns, operators can compare the results with the known scheduled tasks and identify any differences. Collected threat intelligence about unknown scheduled tasks are able to be referenced to identify if any from the results are already associated with malicious activity (Russovich, 2021).

Terrain encompasses which systems are present in the network, including their current configurations and vulnerabilities. Operators can filter analytics to only include analytics used for specific systems that are present in the terrain. For example, if the team has analytics specific to Linux-based techniques, and the terrain doesn't have Linux systems, then the team doesn't have to employ those analytics for that hunt. If a technique is used against a specific type of system or asset, the hunt team can filter the analytic to only include the identified systems or assets. For example, if an adversary is known to target Windows Exchange Server, then the hunt team can tailor the analytics to that server. Filtering the cyber terrain by subnets can also be useful to help focus analytics on smaller areas to help decrease the amount of data an analyst may need to search through at a given time.

In the Cyber Hunt Plan under Filter, add the identified details of time, behavior, and cyber Terrain of systems that may have been compromised. See Table 8 below:

TABLE 8. CYBER HUNT PLAN

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.
Hypotheses and Abstract Analytics	It is suspected that the adversary has used scheduled tasks to establish persistence. CAR analytic <i>CAR-2013-08-001</i> can help hunt for this suspicion.
Determine Data Requirements	Sysmon configuration contains the correct data requirements based on the CAR analytic: instances of schtasks.exe running as processes.
Vulnerability Assessment Results	A vulnerability assessment provides a list of current vulnerabilities that can be used to identify weaknesses the adversary might take advantage of. Knowing which systems are vulnerable to an exploit can help focus the analysis effort of the hunt.
Filter	Time—Filtered based on timeframe of suspected activity Behavior—Filter known good scheduled tasks to identify anomalies Cyber Terrain—Filter by highest priority systems, subnets, or system believed to be compromised

Using Nmap and Zeek to Begin to Filter

Even when the network owner provides a network map/diagram, operators need to validate that the map provided is accurate and identifies rogue systems. The first option is to use Nmap, the network mapping tool, to scan the network for live systems. Operators may be required to run multiple scans using different options to achieve the best results. Table 9 displays some of the basic Nmap commands. Once the scans are complete, compare the results to the network map and update accordingly or identify rogue systems as potential investigation starting points (NMAP, n.d.). Nmap is a robust tool and has many functions to perform a wide variety of scanning techniques. Reference the documentation at <https://nmap.org> for additional functions. While Nmap is not intended to degrade a network it is possible for the tool to degrade networks or crash systems. It is likely that any systems that crash during a scan were already unstable. Operators should review potential legal issues with unauthorized use of the tool and verify if the tool is authorized for use on the network based on organizational policies (Nmap, n.d.). Legal disclaimers and information on potential network impacts can be found at <https://nmap.org/book/legal-issues.html>.

TABLE 9. USING NMAP TO FILTER

Nmap Discovery Options	Command	Example
Scan a single host	<code>nmap [target]</code>	<code>nmap 192.168.0.1</code>
Scan a range of hosts	<code>nmap [range of IP addresses]</code>	<code>nmap 192.168.0.1-254</code>
Ping only scan	<code>nmap -sP [target]</code>	<code>nmap -uP 192.168.0.1-254</code>
TCP Synchronization (SYN) scan	<code>nmap -sS [target]</code>	<code>nmap -sS 192.168.0.1-254</code>
User Datagram Protocol (UDP) scan	<code>nmap -uS [target]</code>	<code>nmap -uS 192.168.0.1-254</code>
Operating System Detection	<code>nmap -O [target]</code>	<code>nmap -O 192.168.0.1-254</code>

The second option to filter the cyber terrain is to use the network data being collected by Zeek (as discussed in the section Zeek on page 70). The Zeek logs identify the common IP Addresses, ports, and protocols found in the network. The Zeek `conn.log` can be used to sort source and destination IPs to identify top talkers and listeners. The challenge

with network data is that it may require several events to provide context into the TTP being investigated. For example, the `conn.log` has fields for `orig_ip_bytes` and `resp_ip_bytes` for a specific event showing the amount of data being transmitted to and from an IP address. The individual event alone may have a small value that may not mean much at that moment but being able to aggregate those fields for multiple events with the same source and destination IP address may indicate potential TTP's around lateral movement, command and control, or data exfiltration if the source and destination IP address are not expected to be communicating at that fidelity. The `conn.log` can also be sorted by service to identify common services and any outliers. There are several network protocol logs with their own respective metadata fields that can provide additional context to the events collected from the network traffic. Understanding the different logs generated by Zeek, what the protocol is designed to do, how that protocol can be abused, and how to leverage the capability of the data analytics tool to view malicious techniques can aid in detecting potentially malicious activity. Reviewing the Zeek documentation will list the additional logs that are generated based on the network traffic being collected for analysis (Zeek, 2021).

Network observation logs from Zeek, such as `known_cert.log`, `known_hosts.log`, `known_services.log`, and `software.log`, can be used to identify abnormal activity that aren't part of the normal baseline. This will also help threat hunt teams to develop a baseline for the network traffic. Hunt teams want to have dashboards that display information such as top-talking IP addresses, HTTP User-Agent, DNS domains queried, and typical port and protocol usage. These dashboards can help identify anomalous activity that will serve as the starting points of an investigation.

Use these dashboards to validate the network map provided and if the specified hosts are operating as expected. Anomalous activity would include hosts acting as a web server when they are not identified as a web server or identifying Service Message Block (SMB) traffic between hosts when there shouldn't be any.

STEP 5: IDENTIFY AND MITIGATE DATA COLLECTION GAPS

STEP 5 COVERS

- Identifying data using Security Onion and its integrated tools
- Improving logging and recommend ways to improve visibility in the network
- Filling data gaps by deploying new sensors

In the previous steps, operators developed abstract analytics based on the intelligence for a malicious activity model and started gaining situational awareness of the network that will be investigated.

When a hunt begins, operators will need to assess which data already exists in the network and determine its visibility. The idea is to determine if the network will have the sufficient data sources of the right quality for detection using ATT&CK techniques. Operators can start by interviewing the network owner to see what they know is already available.

Using the Cyber Hunt Plan (see Table 8 on page 57) as a reference, identify the specific logs or data that are necessary for the analytics in the plan. As operators start collecting data into the SIEM, the data sources being collected can be identified. Security Onion leverages ELK for log ingestion, normalization, and visualizations, which comes with some default dashboards that can be helpful for identifying the data that is being collected. Additional information on the Security Onion tool suite can be found at <https://docs.securityonion.net/en/2.3/pdf/>.

Identifying Data Using Security Onion

The Security Onion tool suite integrates multiple security related tools that allow each of those tools to process collected data. This relieves administrative overhead on security teams having to develop their own custom solutions and allows those teams to focus on threat detection and response. Being able to have these tools integrated into a single solution can allow operators to quickly identify anomalous behavior to drive investigations and security operations. The Security Onion instance comes with several prebuilt dashboards and links to integrated tools to access those tools. Operators start at the navigation dashboard that includes to drill down into the categories *Alerts*, *Hunt*, *PCAP*, and other *tools* that will be outlined in the following sections (see Figure 40 on the following page).

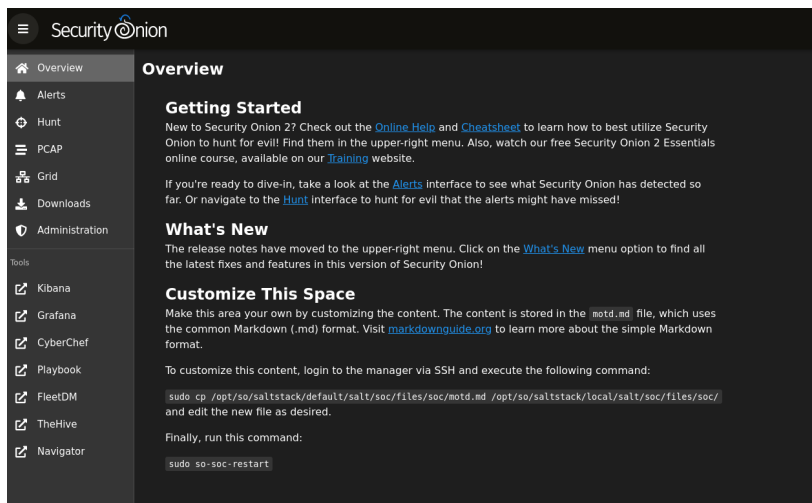


FIGURE 40. SECURITY ONION DASHBOARD

THE ALERTS DASHBOARD DISPLAYS THE ALERT INTERFACE; THIS INTERFACE DISPLAYS THE ALERTS THAT HAVE BEEN GENERATED FROM SURICATA AND OTHER TOOLS.

Alerts

The Alerts dashboard (see Figure 41) displays the Alert interface; this interface displays the alerts that have been generated from Suricata and other tools. The different alerts that are displayed on the screen can either be acknowledge by selecting the bell icon to the right of the alert or by clicking on the blue triangle to escalate the alert. On the top right of interface display shows the amount of time for the alerts, the default is set to 24 hours. The alerts can be used to pivot to other tools, such as PCAP or Hunt, to drill down into the alert to see what future actions will need to be taken.

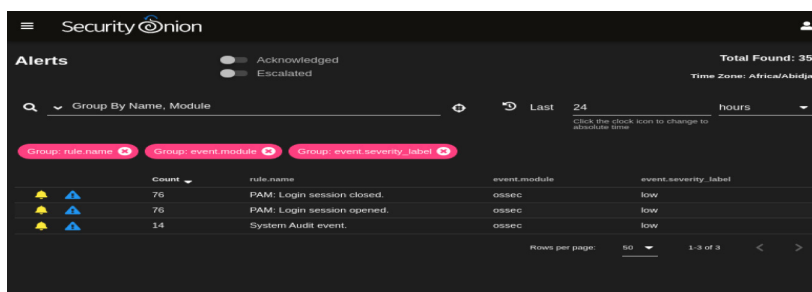


FIGURE 41. SECURITY ONION ALERTS

By clicking on the alert, a new window opens that gives more options in managing the alerts. Figure 42 shows the different options to manage the alert: Include, Exclude, Only, Drilldown, Group by, Hunt, Google, VirusTotal (see Figure 42).

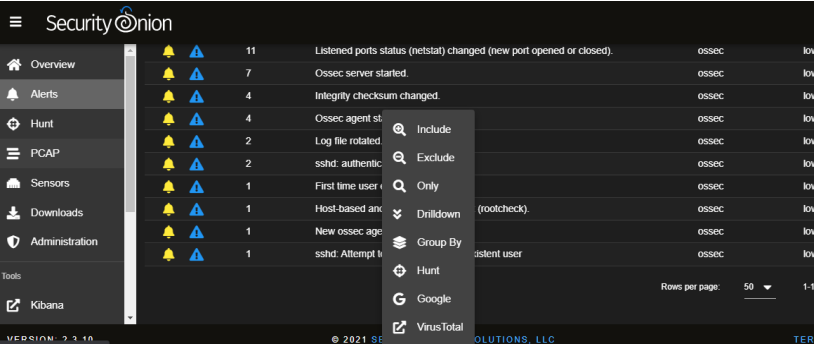


FIGURE 42. ALERT OPTIONS

The Include option will add the value of the alert as a required match in the search, Exclude will remove the value in the search, Only will filter solely on that value. If Drilldown is populated, it will display all the alerts that triggered the event while Group by allows this alert to be grouped with other alerts. The Hunt option will display the alert in the Hunt interface. When the alert is expanded, there may be an option Show PCAP for this event (see Figure 43). By clicking the link, the PCAP for this event will be displayed.

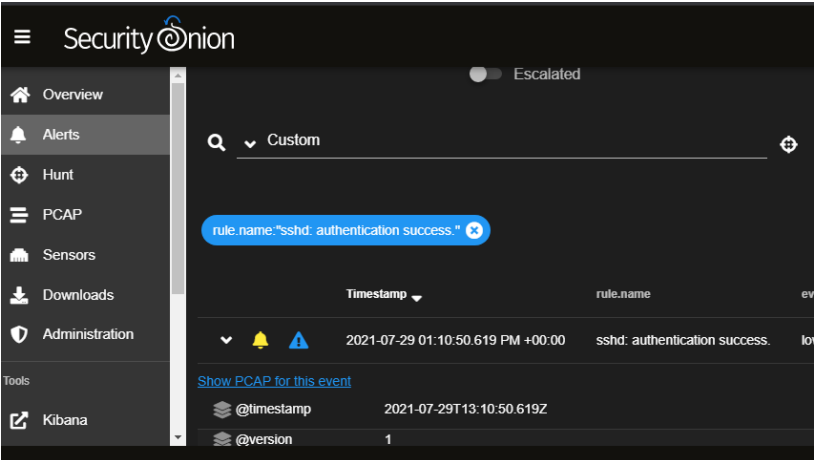


FIGURE 43. ALERT PCAP

Hunt

The Hunt dashboard enables operators to access all of the data within Elasticsearch to perform custom queries (see Figure 44). This interface is tuned for stacking, data expansion, and data reduction techniques. Operators can take the information from the Alerts dashboard and input that data in the Hunt dashboard to investigate those events. Operators should leverage the data developed in the Cyber Hunt Plan and open-source CTI to perform targeted searches on the identified TTP's to test the developed hypothesis.

THE HUNT DASHBOARD
ENABLES OPERATORS
TO ACCESS ALL OF
THE DATA WITHIN
ELASTICSEARCH TO
PERFORM CUSTOM
QUERIES.

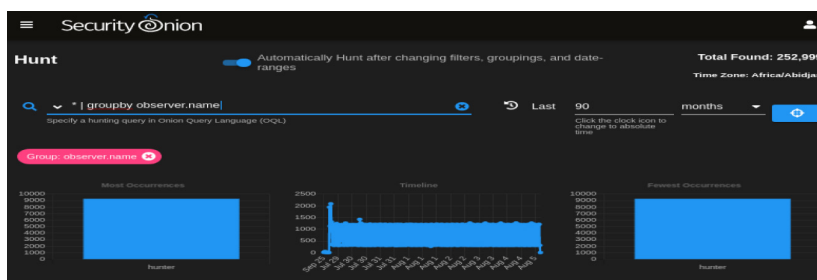


FIGURE 44. SECURITY ONION HUNT

There are several filters and predefined queries with descriptions available to the operator. Queries can also be defined by the time picker allowing operators to search historical data at different time periods if that data is present in the system. There are different visualization and group metrics available to aggregate the data to provide a different context on the events or develop reports (see Figure 45).

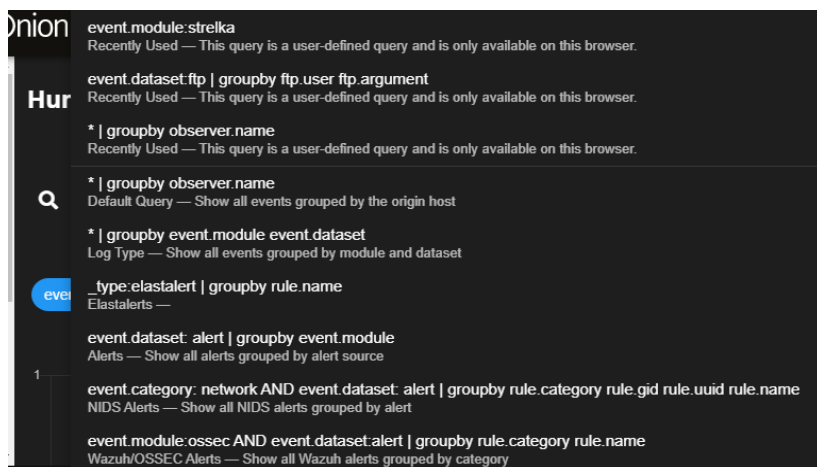


FIGURE 45. SEARCH OPTIONS

THE PCAP INTERFACE SEARCHES FOR THE FULL PCAP CREATED BY STENOGRAPHER.

PCAP

The PCAP interface searches for the full PCAP created by Stenographer. This interface is used to interpret/reassemble network communication captured on the network. Stenographer is a full-packet-capture utility that uses AF-PACKET as the service for the PCAP, this software is used for high performance PCAP writing to disk. Stenographer writes the PCAP in /nsm/pcap/ directory. One method of pulling packet captures is to use the PCAP interface from security onion, and another method is using the stenoread command in the command-line interface or terminal. For more information on how querying Stenographer packet capture, please read <https://github.com/google/stenographer#querying>. PCAP retention and backup policy will need to be taken under consideration. By default, Security Onion will start deleting old data once the partition storage is at 90 percent (Google, 2020).

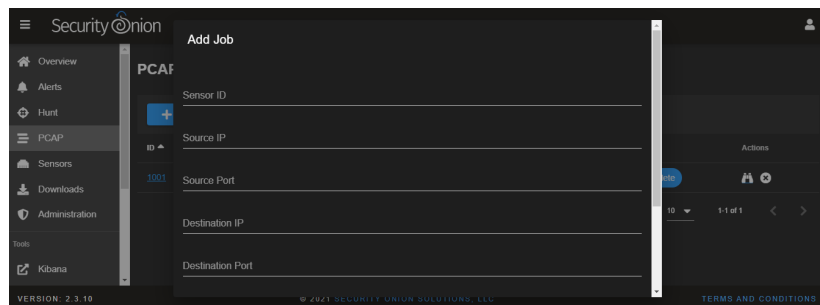


FIGURE 46. PCAP

The value Stenographer adds is by allowing operators to query PCAPs based on the available fields within the dashboard. Some of the fields an operator can search on are Source IP, Source Port, Destination IP, and Destination Port (see Figure 46). Once the operator has entered the required fields with the desired information, the PCAP interface will display the relevant PCAPs to that information (see Figure 47 on the following page).

Stream ID	Timestamp	Protocol	Source IP	Source Port	Destination IP	Destination Port	Length	Offset
0	2016-09-09 03:01:41.254 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	74	SYN
0	2016-09-09 03:01:41.254 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	74	SYN
1	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	66	ACK
1	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	66	ACK
2	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK
2	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK
3	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK
3	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK

FIGURE 47. PCAP STREAM

This shows a high-level view of the network stream. Network streams follow a particular conversation based on the values provided in the fields. If an operator needs to get more granular, they can click on part of the network stream in question to view the PCAP itself instead of the metadata (see Figure 48 below).

Stream ID	Timestamp	Protocol	Source IP	Source Port	Destination IP	Destination Port	Length	Offset
0	2016-09-09 03:01:41.254 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	74	SYN
0	2016-09-09 03:01:41.254 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	74	SYN
1	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	66	ACK
1	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	66	ACK
2	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK
2	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK
3	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK
3	2016-09-09 03:01:41.255 PM +00:00	TCP	173.255.224.88	43100	55.9.19.52	80	1514	ACK

FIGURE 48. PCAP DETAILED

CyberChef

Another tool bundled with Security Onion is CyberChef, which is a web application used for encoding or decoding data (Base64, Hex, binary, etc). For example, if an operator noticed anomalous communication in the PCAP network stream or Sysmon event, that information could be decoded using CyberChef. It's not uncommon for malicious actors to attempt to avoid detection by encoding their commands in Base64.

CYBERCHEF IS A WEB APPLICATION TOOL USED FOR ENCODING OR DECODING DATA.

GRAFANA IS AN OPEN-SOURCE DATA VISUALIZATION TOOL TO DISPLAY DIFFERENT METRICS.

The analyst can copy the data in question and paste it in the input section of CyberChef, then select an operation from the right and move it over the recipe section (see Figure 49). This will convert the data placed in the Input and display in the Output, this is an effective way to see if traffic is being encoded or trying to redirect a system to another site or IP address.



FIGURE 49. SECURITY UNION CYBERCHEF

Grafana

Grafana is an open-source data visualization tool to display different metrics. While Grafana has the capability to display different visualizations based on the data provided, it is used in Security Onion for health metrics. The Grafana link show the systems health information, this can be used to see how much system utilization is happening on the sensor (see Figure 50). Being able to visualize system health metrics can enable operators to address health concerns that may inhibit data ingestion by having a quick visual on memory utilization, Central Processing Unit (CPU) utilization, file system utilization and other data points.

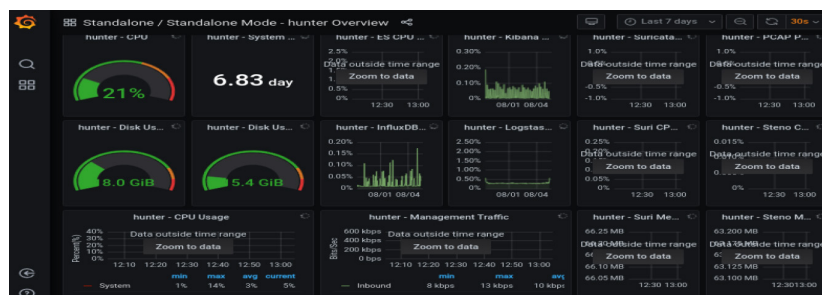


FIGURE 50. SECURITY UNION GRAFANA

TheHive

TheHive is an open-source Security Incident Response Platform designed to enable security operations that has the ability to integrate with other open-source tools. When alerts are escalated from the Alerts interface, the information is populated in TheHive for tracking (see Figure 51 below). TheHive treats the alert as an incident to be tracked and to gather additional information as the alert is investigated. Operators can create task for other operators to help gather additional information and add notes to the incident. This is a useful tool enables collaboration among multiple operators to make sure the alerts are being tracked and managed properly.

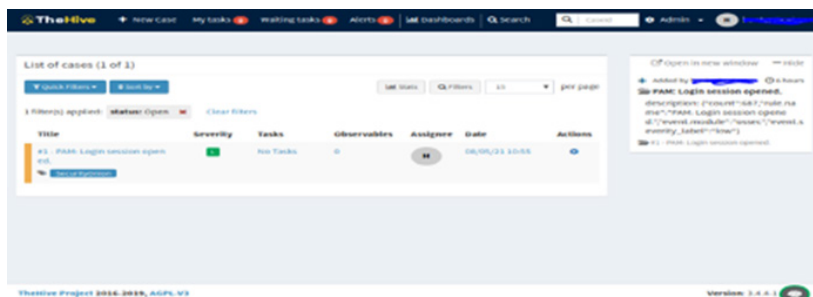


FIGURE 51. SECURITY ONION THEHIVE

Suricata

Suricata is a network threat detection engine that inspects network traffic using rule and signature languages that generates events that will be displayed in Alerts and Hunt interfaces. Suricata started out as a Network Intrusion Detection System and Network Intrusion Prevention System and evolved into an NSM. Suricata can process live network traffic or process PCAP files offline. A Suricata Rules comprises the following, action, header, and rule options (see Figure 52 below).

```
drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"ET TROJAN Likely Bot Nick in IRC (USA +..)"; flow:established,to_server; flowbits:isset,is_proto_irc; content:"NICK "; pcre:"/NICK .*USA.*[0-9]{3,}/i"; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;)
```

FIGURE 52. SURICATA RULE

THEHIVE IS AN OPEN-SOURCE SECURITY INCIDENT RESPONSE PLATFORM DESIGNED TO ENABLE SECURITY OPERATIONS THAT HAS THE ABILITY TO INTEGRATE WITH OTHER OPEN-SOURCE TOOLS.

STRELKA IS A FILE SCANNING SYSTEM THAT SCANS THE FILES EXTRACTED BY ZEEK OR SURICATA AND PERFORMS A RECURSIVE STATIC FILE ANALYSIS.

Figure 52 is the makeup of a Suricata Rule. The part of the rule in red is the action to be taken, and this action is determined when the rule matches. The most common action for network monitoring will be the alert action, this will generate the alert that can be viewed in Security Onion. The part in green is the header, which defines the IP address, port, protocol, and the direction of the rule. In Figure 52, it is using the protocol TCP, moving from \$HOME_NET (i.e., the internal network with any port) to the external network on any port. The protocol option in the headers can use Internet Control Message Protocol (ICMP), UDP, TCP, or IP (which represents all). When discussing the direction of the header (i.e., source and destination traffic), the arrow represented by -> shows the direction. In the example (source -> destination), the source will always be the first and the destination will be after. The direction will be either -> (showing the source packets are moving to on the previous page ward the destination) or <> indicating the packet moving in either direction.

Figure 52 shows the ports as any, which means any port will generate an alert if the rest of the criteria of the rule is met. A specific port can be explicitly stated using the port number and multiple ports can be listed if needed. The \$HOME_NET and the \$EXTERNAL_NET in the Suricata Rule figure can also be represented by the variables used with \$HOME_NET and \$EXTERNAL_NET, have an explicit IP address, or an IP address range with Classless Inter-Domain Routing (CIDR) notation. The parts of the rule in blue are the rule options, which helps to define the rule and will always be enclosed in parentheses. There are additional options available to develop customer rules, which can be found here <https://suricata.readthedocs.io/en/suricata-6.0.3/rules/intro.html>.

Strelka

Strelka is a file scanning system that scans the files extracted by Zeek or Suricata and performs a recursive static file analysis. Once Strelka has scanned these files they are stored in the directory /nsm/strelka/processed/. Strelka can identify over 60 unique files that may be used for malicious purposes. The logs generated by Strelka can be viewed and searched in the Hunt and Kibana interfaces. From the Hunt interface on the search bar, an analyst can select from the drop-down option of type in the event. module:strelka (see Figure 53).

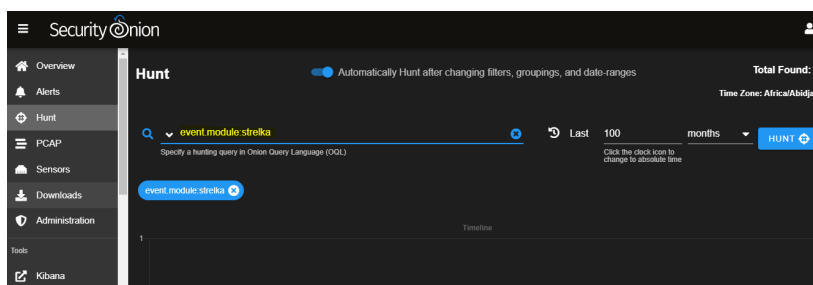


FIGURE 53. STRELKA

Osquery and Fleet

Osquery is an agent that is installed on systems in an infrastructure that makes low-level analytics and monitoring, it creates logs in SQL tables representing abstract concepts as: running process, kernel modules, open network connections, and file hashes. The data that is generated from osquery is in the Hunt or Kibana interfaces by using the search query event.module: osquery. Another way to access osquery is with the Fleet interface as displayed in Figure 54.

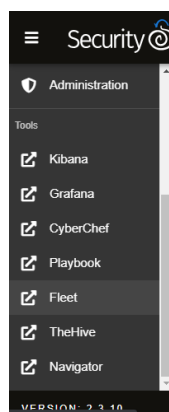


FIGURE 54. FLEET INTERFACE

Fleet is an interface that is used to manage osquery agents, queries, and streaming logs across numerous servers, containers, and hosts, Figure 55 shows the Fleet Interface that can be used to make queries to osquery. Each system that has osquery agents reporting back to Security Onion will be listed under the HOSTS tab.

OSQUERY IS AN AGENT THAT IS INSTALLED ON SYSTEMS IN AN INFRASTRUCTURE THAT MAKES LOW-LEVEL ANALYTICS AND MONITORING, IT CREATES LOGS IN SQL TABLES REPRESENTING ABSTRACT CONCEPTS AS: RUNNING PROCESS, KERNEL MODULES, OPEN NETWORK CONNECTIONS, AND FILE HASHES.

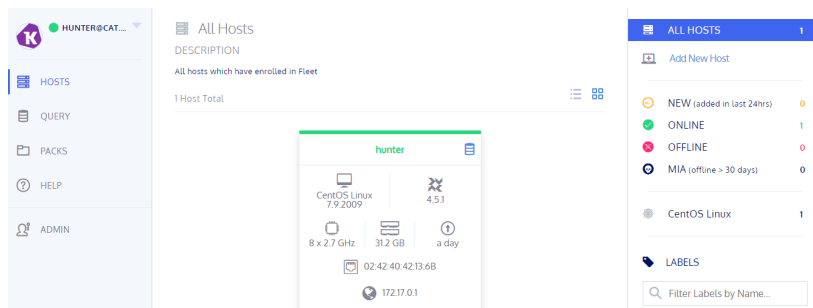


FIGURE 55. FLEET VIEW

Figure 56 shows the Fleet Query tab. If operators are looking for certain types of data about a host, this is the location to make that query. Operators can schedule queries to be executed across the endpoints with the agent installed on to retrieve information. The types of information that can be retrieved are running processes, user logins, loaded kernel modules, open network connections, browser plugins, hardware events, file hashes, and more.

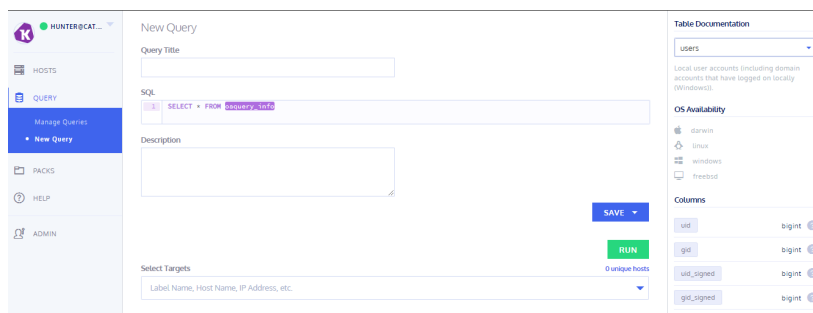


FIGURE 56. FLEET QUERY

Zeek

Security Onion tool suite leverages Zeek, formerly called Bro, as a passive NSM working as a traffic analyzer that generates metadata on the analyzed traffic. This metadata is outputted to different logs to be ingested into the ELK stack to enable operators to perform queries on that data to potentially identify TTPs and malicious behaviors. That data is collected through the promiscuous

port, also referred as the Sniffing NIC, and sent to the AF_PACKET service to receive the raw packet. That information is sent to multiple tools and is received by Zeek to perform its traffic analysis (see Figure 57) (Zeek, n.d.).

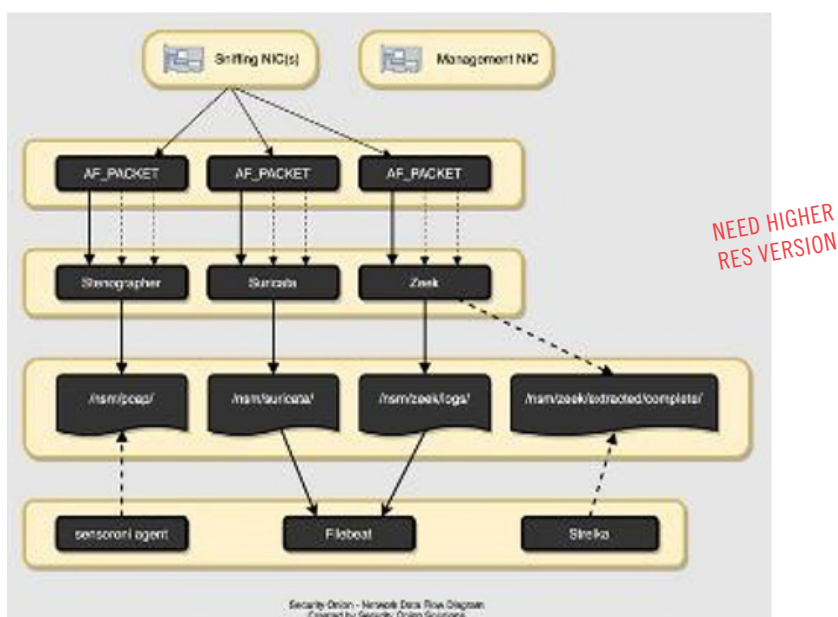


FIGURE 57. SECURITY ONION 2 NETWORK FLOW DIAGRAM

Zeek will run the collected data through its own Event Engine, NetControl, and Policy Script Interpreter to create logs that are stored in `/nsm/zeek/logs/`. Zeek will make a copies of files identified in the data that are being transferred on network and tags that data as an extracted file. Extracted files are stored in the directory `/nsm/zeek/extracted/complete/`, which operators can retrieve for further analysis. Keep in mind that some of these files may be malicious and executing them in an unsecure location can infect the analysis system. Zeek can monitor live network traffic and process saved PCAP files to generate logs. To see what default Zeek scripts are running on Security Onion, the command `so-zeek-logs` can be run from a terminal connection to the sensor that will display a list of scripts running to generate logs, Once the command is run, the terminal will output a list of log sources with a brief description of the logs.

The proper Zeek script is running to generate the respective log if an asterisk is present between the square brackets (see Figure 58).

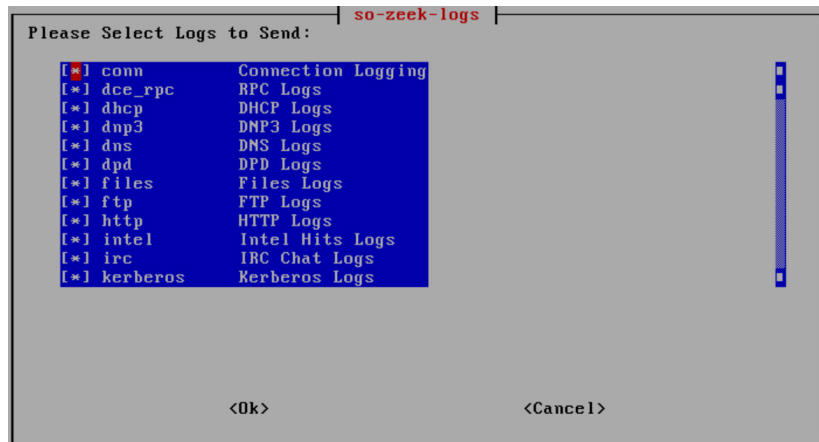


FIGURE 58. SO-ZEEK-LOGS

While administrators can select specific logs, there are several default logs that come enabled. The log files are stored in the `/nsm/zeek/logs` directly and ingested into the ELK stack with Filebeats. The naming convention of the files indicate the type of data being collected and stored within the logs (see Figure 59). That data is then easily accessible through Kibana and the Hunt dashboard for operators to investigate.

Zeek Logs

- `conn.log`
- `dns.log`
- `http.log`
- `files.log`
- `ftp.log`
- `ssl.log`
- `x509.log`
- `smtp.log`
- `ssh.log`
- `pe.log`
- `dhcp.log`
- `ntp.log`
- `SMB Logs (plus DCE-RPC, Kerberos, NTLM)`
- `irc.log`
- `rdp.log`
- `traceroute.log`
- `tunnel.log`
- `dpd.log`
- `known_*.log` and `software.log`
- `weird.log` and `notice.log`
- `capture_loss.log` and `reporter.log`

FIGURE 59. EXAMPLES OF ZEEK LOGS

An example of the data being analyzed by Zeek is the conn.log. The conn.log script tracks the ICMP, TCP, and UDP traffic and generates several fields that operators can filter on. This log shows which endpoint is connecting to other devices with related fields with duration of the connections, protocols identified, the connection state, source IP address, destination IP address, and many more (see Figure 60). Operators should become familiar with all the logs available and the respective metadata fields unique to those log types to develop effective analytics. Additional information on the logs and their respective fields can be found in the Security Onion or Zeek documentation.



```

File Edit Search View Document Help
/run/user/1000/gvfs/sftp:host=192.168.3.176/nsm/zeek/logs/current/conn.log - Mousepad
{"ts": "2021-07-29T12:59:56.963222Z", "uid": "CPgeCS1MfREUe7U4n", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:00:37.372681Z", "uid": "CmYDQg42Elmpriyhve", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:01:56.959307Z", "uid": "Cy2rVe2GGzMQ0fHmag", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:02:37.373662Z", "uid": "C9nzoH1fHvKbTzgm9l", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:03:44.509327Z", "uid": "CtSdtb1DWdsPOwtsfj", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:03:44.508365Z", "uid": "C98ALW21SSCuIOkct9", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:03:44.499432Z", "uid": "Cnp2Ln10yCXETDp5u4", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:03:44.504940Z", "uid": "CAphk821x8Cu4zaXl1", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:03:56.959437Z", "uid": "CcQZrG2LP5HT90SeIf", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:04:37.374515Z", "uid": "CX5Qgk4cf2x318Hjhj", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:05:56.960119Z", "uid": "CvLtvV2ov83Xymvbs", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:06:37.373293Z", "uid": "CxI7j447dJoluMaN6b", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:07:56.961589Z", "uid": "CKM9M9k1mL5SeADV9Ae", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:08:37.373879Z", "uid": "C88xxu1pFy5rN3a8", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:09:56.961943Z", "uid": "CrU5Sn4GhJmDTPXMB1", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:10:37.374880Z", "uid": "C8Hf0J1K400j4Mfcc", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:11:56.963378Z", "uid": "Cum7dt0xm12lVKtng", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:12:37.375173Z", "uid": "CjkVMENCWPYIRgKw5", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:13:56.964045Z", "uid": "CoT9v23uxgthBWN7R3", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:14:37.375242Z", "uid": "C9LSjyYACl9Injuv8", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:15:56.965591Z", "uid": "CFS15a1He4s8J2u3P2", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:17:29.401094Z", "uid": "CtNW394dmeOyMx5JN1", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:16:37.375314Z", "uid": "C9dZTy493IOW3cRcE1", "id.orig_h": "192.168.159.1", "id.orig_p": 4444, "id.resp_h": "192.168.3.175", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:44:48.471968Z", "uid": "C3r5XxufCcNex6aub", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}
{"ts": "2021-07-29T13:17:59.852998Z", "uid": "Ce9EUE3Vgd7lkEnP6", "id.orig_h": "192.168.3.175", "id.orig_p": 4444, "id.resp_h": "192.168.159.1", "id.resp_p": 4444, "proto": "TCP", "state": "ESTABLISHED", "duration": 0.0, "len": 0, "info": ""}

```

FIGURE 60. EXAMPLE ZEEK CONN.LOG

One limitation of this script is that it will only log the connections after the event has finished. Until then, it will not log any information. Zeek also has a 5-minute timeout. If the connection is not established in the 5 minutes allotted, it will purge and not log. The timeout can be extended by modifying the tcp_inactivity_timeout in local.zeek file, and depending on system utilization, the time can be extended to 30 or 60 minutes.

When operators must run Zeek manually against PCAP, it will need to be done from a terminal in a new directory for the logs it will generate. The command to execute is zeek -C -r and the PCAP file in question: the -r command tells zeek which PCAP to read, and the -c ignores checksum errors. Running this command will generate the logs from Figure 60. If the log files present too much data, the logs will wrap the output, making the log difficult to read. The zeek-cut command can help make the output more presentable, knowing the desired fields can create the desired output. For example, if an operator wanted information about the communication between two different systems and what ports were being used, the following command would be used:

```
cat conn.log | zeek-cut id.orig_h id.orig_p id.resp_h id.resp_p
```

KIBANA IS A DATA ANALYTICS TOOL THAT ALLOWS OPERATORS TO VISUALIZE THE DATA BEING COLLECTED ON A THREAT HUNT ENGAGEMENT.

Here's a breakdown of several of facets of the above command (Zeek, 2021):

- `id.orig_h` is the originating endpoint IP address
- `id.orig_p` originating endpoint port used for connection
- `id.resp_h` is the responding endpoint IP address
- `id.resp_p` is responding endpoint port being used

Kibana

Kibana is a data analytics tool that allows operators to visualize the data being collected on a threat hunt engagement. When operators first access Kibana, several metrics will be displayed to convey quick information on the collected data. This section of the dashboard includes links to drill down into the categories *Alert*, *File*, *Host*, and *Network*, all of which provide data (see Figure 61).

- **Alert:** Information on Suricata and Playbook alerts
- **File:** Information on files that have been extracted from Zeek and processed in Strelka
- **Host:** Information from hosts with osquery, Event Logs, and Sysmon
- **Network:** Information for network sensors, such as Zeek

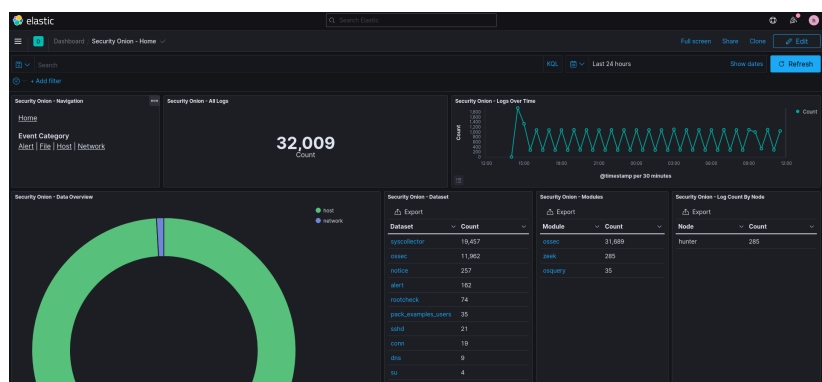
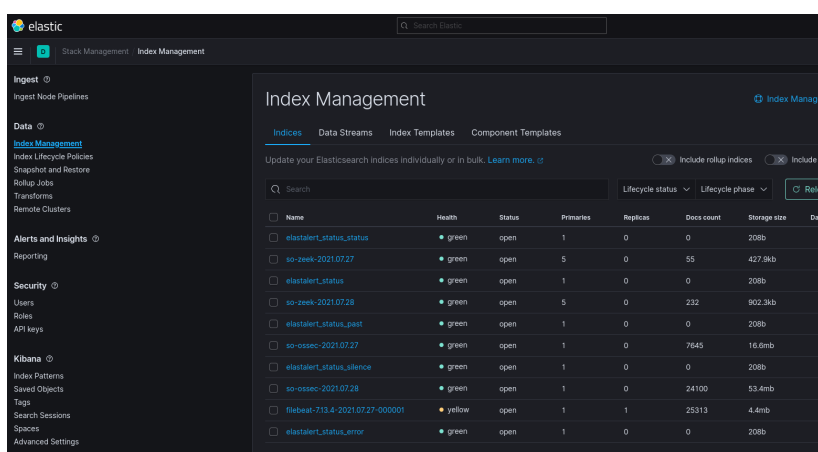


FIGURE 61. KIBANA MAIN PAGE

The figure displays the Kibana data visualization interface for Elasticsearch that allows querying data in the Elastic Stack. Data that is collected from host-based sensor and network-based sensor is forwarded to the ELK Stack to be searchable and able to provide analytics on this information. Kibana and Elasticsearch will be accessible from a web interface. Operators will need to know the URL and have an account setup for access.

Elasticsearch indexes are tied to a certain data source which Security Onion has already defined for the data it collects. Examples of data sources are Zeek logs, Suricata logs, host logs, and data generated from the tools mentioned above. Additional log sources can be forwarded to the ELK stack and operators can create custom indexes to organize the data and set custom retention periods. To understand what is being indexed to Elasticsearch, on the Elasticsearch homepage, click the menu tap, 3 horizontal lines, and go to the Management section and click Stack Management (see Figure 62). Then, click Index Management, which will display the Elasticsearch indices that are searchable from Kibana. Navigate to the Discover page in Kibana, and the search bar will be available. The Discover page allows operators to search by the indices and available fields and setting search parameters by date and time. Each dataset represents a log with its name as a link that allows the operator to drill down into that specified dataset.



The screenshot shows the Kibana Index Management page. The left sidebar contains navigation links for Ingest, Data, Alerts and Insights, Security, and Kibana. The main content area is titled 'Index Management' and includes tabs for Indices, Data Streams, Index Templates, and Component Templates. Below the tabs, there are filters for 'Include rollout indices' and 'Include hidden indices'. A search bar is present above a table of indices. The table has columns for Name, Health, Status, Primary, Replicas, Docs count, Storage size, and Date.

Name	Health	Status	Primary	Replicas	Docs count	Storage size	Date
elasticsearch_status	green	open	1	0	0	208b	
so-zeek-2021.07.27	green	open	5	0	55	427.8kb	
elasticsearch_status	green	open	1	0	0	208b	
so-zeek-2021.07.28	green	open	5	0	232	902.3kb	
elasticsearch_status_past	green	open	1	0	0	208b	
so-ossec-2021.07.27	green	open	1	0	7645	16.6mb	
elasticsearch_status_silence	green	open	1	0	0	208b	
so-ossec-2021.07.28	green	open	1	0	24100	53.4mb	
filebeat-713.4-2021.07.27-000001	yellow	open	1	1	25313	4.4mb	
elasticsearch_status_error	green	open	1	0	0	208b	

FIGURE 62. INDEX MANAGEMENT

Relating Tools to the Cyber Hunt Plan

Now that a basic overview has been given about Security Onions different tools, it's time to focus on how to use these tools to identify malicious activity in relation to the Cyber Hunt Plan being developed. All of the mentioned tools allow operators to filter the collected data to perform targeted searches with the developed analytics. The Cyber Hunt Plan focuses on the Execution Tactic of Scheduled Task T1053.005, which will be Windows Event ID 4698 "A scheduled task was created." If all Windows logs are ingested into Elastic Stack, then the following query can be used to search for these events: *winlog.event_id:"4698"*; this will show every system that has Event ID 4698. If operators are looking for this ID on a certain system, the *winlog.EventData.DisplayName* and the system name. For example:

(winlog.EventData.DisplayName:"Test_Host" AND winlog.event_id:"4698").

Sysmon is also able to log system information. If Sysmon is being ingest instead of Windows event logs, the *winlog.channel* can be queried with Microsoft-Windows-Sysmon/Operation or Sysmon and the Event ID 1 "Process creation." For example:

(winlog.channel:"Sysmon" AND winlog.event_id:"1").

The *winlog.channel* can also be used to point to different Windows Event Logs, such as Security, Application, System, Setup, and Sysmon. If the result of the query above shows a command line connection to an IP address in *winlog.event_data.CommandLine*, leveraging another data source like Zeek *conn.log* can provide valuable information with connections to the identified IP address. To search Zeek logs, the *event.module* and *destination.ip* are the needed filters to use in the Kibana search field, for example:

(event.module:"zeek" AND destination.ip:"ip_address_to_look_up")

This query will show how often the system is reaching out to an identified IP address, then check the *event.dataset* to see what type of protocols are being used during this connection. The Sysmon Event ID 1 will provide the command line used to create the scheduled tasks under the field *winlog.event_data.CommandLine*. PowerShell has the capability to encode commands used in Base64 to potentially avoid detection. This is where CyberChef can be used to decode the Base64 string and find out what information the attacker is trying to hide, refer to the section on CyberChef on page 65. Another consideration with Sysmon is if a file is created, deleted, or transferred to a system it will log the hash of the file. Identified hashes can potentially be used to identify known files or software that may be used maliciously by entering the hash into an OSINT tool or site.

Improving Logging and Recommended Visibility of the Network

After operators have identified the data that needs to be collected is being ingested and processed properly, they must assess the validity of the data. One method to check that the data is present is a simple frequency analysis of relevant event codes over time to detect periods when collection of that event may have been disrupted. Another way to perform a validity check is to compare results from different data sources to ensure consistency. Operators can use frequency analysis of event counts by IP address or hostname to identify coverage gaps across the terrain.

Appendix C provides the MITRE ATT&CK TTP heat map of APT29. This table maps TTPs to the ADOS tools can detect that technique. Now, let's cover how these tools can be used to detect malicious activity. On the APT29 table under Execution, Command and Scripting Interpreter, PowerShell T1059.001 can be detected from host-based data, meaning the event logs or Sysmon. Sysmon must be installed and configured on each host device and event logging needs to be configured as well to capture the correct data. The Windows Event logs can be viewed from the Windows Event Viewer. If logging is set correctly, searching for Event ID 400 or 600 will show information about PowerShell scripts or commands that have been run on that machine. If looking at Sysmon directly on the host, operators will need to search for multiple events as Sysmon does not have an Event ID specifically for PowerShell. An example is Sysmon Event ID 1 process creation, Sysmon Event ID 11 FileCreate, and Sysmon Event ID 15 FileCreateHash. These events used together can help detect if a PowerShell command or script has run on a system depending on PowerShell actions. It is helpful to know what logs provide the most data about the host that is associated with Events.

Filling Data Gaps by Deploying New Sensors

Networks are frequently missing the data sources required to identify malicious cyber activity. Operators should be prepared to deploy new sensors within the network. Having step-by-step guides for each sensor can help to expedite this process.

Refer to the section "Deploy Sensors" on page 10 regarding the potential impact introducing new sensors and applications may have to the network. The same practices apply with testing any enhancements to current data

sensors or adding new sensors on a small subset of systems to determine impacts on the systems and generate new baselines. Operators should ensure the new sensors don't affect the normal functions of the systems, such as causing spikes in memory or processor usage. Operators should monitor the increased levels of logging to ensure that there will be enough storage to capture the logs and that the network can handle the increased network traffic from the additional logs. Beware that several security vendor products use similar methods to collect data and might conflict with each other. Operators must verify the new sensors are not disrupting collection from existing sensors.

In the event the desired sensor or sensor enhancements are not able to be installed, operators should identify compensating controls to collect the required data. For example, operators may only be able to collect network traffic and must rely on forensic analysis of a system to obtain the host data. Bear in mind that forensic analysis is conducted on an as-needed basis, and therefore it does not provide the same kind of continuous monitoring provided by log collection.

TABLE 10. CYBER HUNT PLAN

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.
Hypotheses and Abstract Analytics	It is suspected that the adversary has used scheduled tasks to establish persistence. CAR analytic <i>CAR-2013-08-001</i> can help hunt for this suspicion.
Determine Data Requirements	Sysmon configuration contains the correct data requirements based on the CAR analytic: instances of schtasks.exe running as processes.
Vulnerability Assessment Results	A vulnerability assessment provides a list of current vulnerabilities that can be used to identify weaknesses the adversary might take advantage of. Knowing which systems are vulnerable to an exploit can help focus the analysis effort of the hunt.
Filter	Time—Filtered based on timeframe of suspected activity. Behavior—Filter known good scheduled tasks to identify anomalies. Cyber Terrain—Filter by highest priority systems, subnets, or system believed to be compromised.
Find and Mitigate Data Collection Gaps	Identify data sources that are available and annotate. Document data sources that aren't available and recommend logging improvements. Install Sysmon. Install and execute Autoruns.

STEP 6: IMPLEMENT AND TEST ANALYTICS

Implementing Pseudocode Analytic to Kibana

This section will describe how operators can implement a pseudocode analytic developed previously and translate it into Kibana Query Language (KQL) for use within Kibana. The analytic will then be tested using threat emulation to generate the required data for logging to occur.

Using the identified analytic *CAR-2013-08-001: Execution with schtasks* from Step 2, operators will need translate the pseudocode analytic into a query that can be used within Kibana. To review, the purpose of this analytic is to find instances of schtasks.exe running as a process. The additional flags will be added as arguments to the Kibana query.

```
process = search Process:Create
schtasks = filter process where (exe == "schtasks.exe")
output schtasks
```

FIGURE 63. CAR-2013-08-001: EXECUTION WITH SCHEDULED TASKS

Kibana uses the query languages KQL and Lucene. KQL is the syntax for filtering Elasticsearch data using free text search or field-based search. KQL does not support regular expression (regex) or searching with fuzzy terms while Lucene supports those search functions. Regular Expression (Regex) or fuzzy searches will impact search performance as they require more resources to perform. However, Lucene is not able to search nested objects or scripted fields. KQL will be the most common query language operators will use, but Lucene may have its use cases and should be considered depending on the situation (Elastic, n.d.). Elastic uses the Elastic Common Schema (ECS) to define a common set of fields for storing event data. To develop the KQL query, identify the appropriate ECS fields to use in the query to filter on different attributes within the event. Using the ECS Field Reference, operators can find the section on Process Fields, which contains the fields about a process (Elastic, n.d.). For this example, operators will use process.name to identify the names of the process in the query and process.args to identify the possible arguments used with command in order to build the following query. This query will be added to your Cyber Hunt Plan as shown in Table 11 on page 82.

```
process.name:schtasks.exe and process.args:("/create" or "-create" or "/S" or "-s" or "/run" or "/change" or "-change")
```

FIGURE 64. KIBANA QUERY FOR SCHEDULED TASKS

STEP 6 COVERS

- Implementing pseudocode analytic to Kibana
- Testing analytics
- Exploring adversary emulation

OPERATORS NEED TO HAVE A CLEAR UNDERSTANDING OF THE DATA TYPES FEEDING INTO THE SOLUTION TO BUILD EFFECTIVE ANALYTICS.

Elastic provides prebuilt rules and additional guidance and examples for queries to use within Elastic. The rules can be found at <https://www.elastic.co/guide/en/security/current/prebuilt-rules.html>.

Testing Analytics

To verify that this analytic is ready for operation, it should be tested for validity, precision, recall, and performance. An analytic is valid if it accurately represents the logic of the hypothesis. This means the syntax must be correct, and the logic properly implemented. An analytic has perfect precision if it does not generate any false positives. To test the precision of an analytic, the threat hunt team can run it over a long window of time across the full terrain of the target network and count how many false positive results are returned. Analytics with too many false positives are likely to be ignored by analysts and lose their value for detection. Such analytics should be modified to reduce the false positives or reserved for more forensic use cases. An analytic has perfect recall if it detects all instances of the malicious behavior. This can be validated by emulating the behavior in the target network to check that it is detected. The behavior should be emulated in several different ways to test the robustness of the analytic to different procedural implementations. This process is often called “purple teaming.” An analytic should also have good performance, returning results in a reasonable amount of time. No analytic will have perfect precision, recall, and performance. Acceptable thresholds of precision, recall, and performance will be dependent on the situation. With sufficient precision, recall, and performance across a wide range of emulations of the behavior and across the full target network, the analytic has a solid foundation for continuous operations.

Operators need to have a clear understanding of the data types feeding into the solution to build effective analytics. This would allow operators to optimize the performance of the SIEM by targeting effective data sources that may provide additional fields to enhance the effectiveness of the analytic. For example, operators can use the Zeek conn.log to perform a search on port 3389 to detect Remote Desktop Protocol (RDP) connections. The event will allow operators to determine if an RDP connection was successful, duration of the connection, number of bytes transferred, and the IP addresses of the connection. Pairing this information with the Zeek rdp.log can provide additional information such as cookie, security protocol, client name, and encryption level which may be valuable depending on the use case.

There are many data types based on the tool generating the data and the protocols being analyzed with their own respective fields. Different data types may even have the same field name but may not have the same meaning in context of the protocol or event being analyzed. Developing a data dictionary can aid in understanding the different data types and the corresponding field names. The Open Threat Research Forge (OTRF) Open-Source Security Events Metadata (OSSEM) project is a community driven project aimed at documenting and standardizing security event logs from diverse data sources. The OSSEM project aligns with MITRE ATT&CK and is a good resource for reviewing event mappings to MITRE ATT&CK TTP's, additional data sources, and event IDs related to the technique provided (Rodriguez, 2020).

Exploring Adversary Emulation

Adversary emulation is a form of an offensive engagement (i.e., red teaming) that emulates a known threat or technique on a network or subset of systems. The goal of conducting adversary emulation is to identify detection gaps in the host-based and network-based sensors on specific techniques. This will aid in the maturation of the security controls by either enhancing the logging capabilities or allow organizations to make data driven decisions with incorporating new security tools that would generate the required telemetry. Adversary emulation can also be used to develop or enhance analytics to detect on specific techniques that are known by specific threat actors targeting the organization or identified in the Cyber Hunt Plan. Adversary emulation is meant to mimic real attacks used by threat actors and should not be performed on production systems. Instead, a lab or development environment that mimics the production environment should be used. To properly test analytics using adversary emulation, the same sensors and security controls must be in place on the host and inline between the attacker and victim host as they would be on the production environment.

MITRE's Caldera™ tool is an open-source, autonomous adversary emulation tool. Caldera is built on the MITRE ATT&CK framework and consists of two main components: an asynchronous C2 server and plugins to provide additional functionality. Users can select specific ATT&CK techniques (referred to as abilities within Caldera) or run adversary profiles that consist of a collection of abilities of the selected adversary. Running the abilities or adversary profile against the target system is referred to as an "operation," which users can save as an "operation planner." Additional functionality

can be added to Caldera through the plugin library. There are several plugins available that can mimic end user behavior: shell and reverse shell functionality, integrated Atomic Red Team TTPs, and training for the end user. Additional information on Caldera, plugins, and emulation plans can be found at (MITRE, 2021):

- <https://caldera.readthedocs.io/en/latest/Installing-CALDERA.html>
- <https://caldera.readthedocs.io/en/latest/Plugin-library.html>
- https://github.com/mitre-attack/attack-arsenal/tree/master/adversary_emulation/APT29
- https://github.com/center-for-threat-informed-defense/adversary_emulation_library

TABLE 11. CYBER HUNT PLAN

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.
Hypotheses and Abstract Analytics	It is suspected that the adversary has used scheduled tasks to establish persistence. CAR analytic <i>CAR-2013-08-001</i> can help hunt for this suspicion.
Determine Data Requirements	Sysmon configuration contains the correct data requirements based on the CAR analytic: instances of schtasks.exe running as processes.
Vulnerability Assessment Results	A vulnerability assessment provides a list of current vulnerabilities that can be used to identify weaknesses the adversary might take advantage of. Knowing which systems are vulnerable to an exploit can help focus the analysis effort of the hunt.
Filter	Time—Filtered based on timeframe of suspected activity. Behavior—Filter known good scheduled tasks to identify anomalies. Cyber Terrain—Filter by highest priority systems, subnets, or system believed to be compromised.
Find and Mitigate Data Collection Gaps	Identify data sources that are available and annotate. Document data sources that aren't available and recommend logging improvements. Install Sysmon. Install and execute Autoruns.
Implement and Test Analytics	Execute the following KQL query in Kibana. process.name:schtasks.exe and process.args:("/create" or "-create" or "/S" or "-s" or "/run" or "/change" or "-change")

STEP 7: HUNT/DETECT MALICIOUS ACTIVITY AND INVESTIGATE

Locating Threat Hunting in the Incident Response Process

In the previous steps, it was determined which adversary to model the hunt operations around, and a malicious activity model was developed around that threat. A pseudocode analytic was developed to hunt for a specific technique used by the adversary in the malicious activity model. Available data sources and collection gaps were identified in the system. Next, the pseudocode analytic was converted into a KQL query that could be used within Kibana. Now operators are going to operationalize the analytic and begin hunting.

Hunting is an interactive process that requires creativity and flexibility. Hunting will usually fall into the Detection & Analysis step of the Incident Response Lifecycle (Cichonski, Millar, Grance, & Scarfone, 2012) (see Figure 65). There are variations of incident response processes with different steps, but hunting is considered to fall within the step that covers detection and analysis of malicious activity.

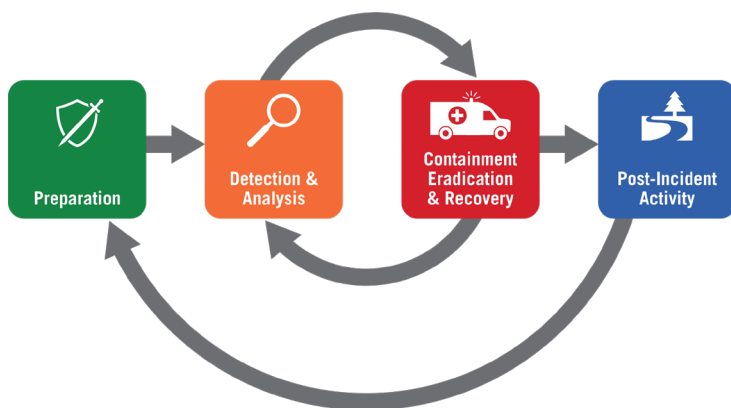


FIGURE 65. NIST INCIDENT RESPONSE LIFECYCLE

Hunting is enabled by a core sequence of steps. It begins with collected data and knowledge of malicious TTPs and builds on existing knowledge to filter the data efficiently and find malicious activity. Once the malicious activity is sufficiently understood, the organization can implement containment and eradication procedures to impose cost on the adversary.

STEP 7 COVERS

- Locating hunting in an overall Incident Response process
- Tuning the analytic(s) for initial detection
- Evaluating the Events
- Documenting the malicious events
- Gathering contextual information
- Investigating the malicious events
- Concluding the cyber hunt plan
- Responding to the security incident
- Assessing analytics and hunt processes
- Reviewing additional considerations for the threat hunting process

Each step in the process is described in greater detail in the following sections.

Many cybersecurity professionals are familiar with Lockheed Martin's Cyber Kill Chain intelligence driven defense model. The Cyber Kill Chain identifies the steps that adversaries must complete to complete their goals. The goal of the model is to enable security teams to identify and stop attackers at every stage of the kill chain. The seven stages of the Cyber Kill Chain are:

1. **Reconnaissance:** The attacker collects information on the potential victim through, but not limited to, social engineering, open-source intelligence, organization web pages and postings, and scanning techniques to prepare an attack on the target.
2. **Weaponization:** Attackers develop the tools and techniques that will be used to potentially compromise the intended target. Attackers do not interact with the potential victim.
3. **Delivery:** Built on the previous stages, Attackers transmit their attacks to gain initial access and set up persistence on the victim. This can be achieved through different techniques but is commonly seen with phishing attempts and social engineering.
4. **Exploitation:** The attack that was successfully delivered to the victim is activated running the exploit on the compromised systems.
5. **Installation:** The attacker installs malware on the victim system.
6. **Command and Control:** Once the system is compromised, it calls home to a Command-and-Control system for the attacker to gain control.
7. **Actions on Objective:** The attacker has established access on the victim's network and can execute actions to achieve their objectives.

Threat hunting engagements take place between the delivery and the actions on objective stage as the attacker is interacting with the intended victim. Operators won't be able to prevent attackers from completing the reconnaissance or weaponization stages as there is no direct interaction with the internal infrastructure of the network at that point (Hutchins, Cloppert, & Amin, n.d.).

It is important to be aware of different industry lifecycles or frameworks to aid security teams to develop effective processes unique to their use case. The TTP-Based Hunting Methodology and developing a Cyber Hunt Plan is comprehensive and can aid security teams in maturing their processes to potentially detect malicious activity (see Figure 66).

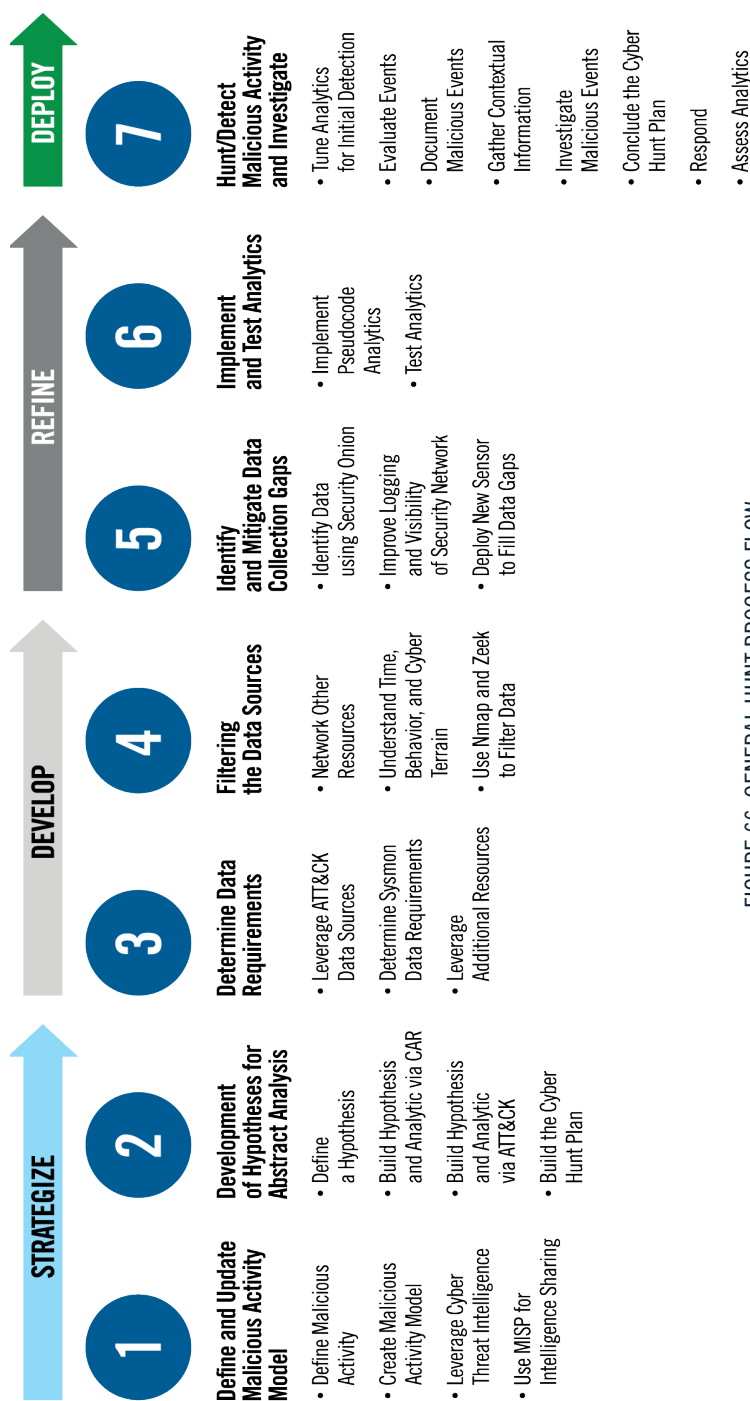


FIGURE 66. GENERAL HUNT PROCESS FLOW

Tuning Analytic(s) for Initial Detection

When the hunt begins, operators will need to tune the analytic to identify malicious events accurately. Narrowing the space across which results are queried will reduce the total number of events to be analyzed. This is a tradeoff as broadening the results may also reveal patterns that would go unnoticed in a narrow time window. Table 12 lists four primary techniques as defined in Hunt Evil: Your Practical Guide to Threat Hunting (Sqrrl, n.d.).

TABLE 12. THREAT HUNTING TECHNIQUES

Four Primary Hunting Techniques	
Searching	The simplest method of hunting, searching is the process of querying data for specific results or artifacts and can be performed using many tools. Searching requires finely defined search criteria to prevent result overload. There are two primary factors to keep in mind when carrying out a search: searching too broadly for general artifacts may produce far too many results to be useful and searching too specifically for artifacts on specific hosts may produce fewer results than may be useful.
Clustering	Clustering is a statistical technique, often carried out with machine learning, which consists of separating groups (or clusters) of similar data points based on certain characteristics out of a larger set of data. Hunters may use clustering for many applications, including outlier detection, due to the fact that it can accurately find aggregate behaviors, such as an uncommon number of instances of a certain occurrence. This technique is most effective when dealing with a large group of data points that do not explicitly share immediately obvious behavioral characteristics.
Grouping	Grouping consists of taking a set of multiple unique artifacts and identifying when multiple of them appear together based on specific criteria. The major difference between grouping and clustering is that in grouping the input is an explicit set of items that are already of interest. Discovered groups within these items of interest may potentially represent a tool or a TTP that an attacker might be using. An important aspect of using this technique consists of determining the specific criteria used to group the items, such as events having occurred during a specific time window. This technique works best when hunting for multiple, related instances of unique artifacts, such as the case of isolating reconnaissance commands that were executed within a specific timeframe.
Stack Counting	Also known as stacking, this is one of the most common techniques carried out by hunters to investigate a hypothesis. Stacking involves counting the number of occurrences for values of a particular type and analyzing the outliers or extremes of those results. The effectiveness of this technique is generally diminished when dealing with large and/or diverse data sets, but it is most effective with a thoughtfully filtered input (such as endpoints of a similar function, organizational unit, etc.). Analysts should attempt to understand input well enough to predict the volume of the output. For example, if given a dataset containing 100k endpoints, stack counting the contents of the Windows\Temp\ folder on each endpoint across an enterprise will produce an enormous result set. Friendly intelligence can be used to define filters for the input.

An example of a technique that can be detected using the hunting techniques listed above is a brute force attack. A brute force attack is when an attacker attempts every possible combination of the targeted password until the account is compromised. Using the search technique, operators would use the Kibana interface to query for accounts with failed logins. If operators are successfully collecting Windows Event Logs, the query should focus on Event ID 4625 for failed logon attempts followed by the Event ID 4624 logon successful. It's not uncommon for users to fail login attempts before successfully authenticating. The operator can use the stack counting technique to look at the number of failed login attempts. A large number of failed login attempts may indicate a brute force attack is being attempted, and if there's a successful logon after numerous failed login attempts, it may indicate that a brute force attack was successful in compromising an account. This information could help operators pivot their searches to identify additional TTP's if the attacker attempted privilege escalation or lateral movement from that compromised account. This can also be an opportunity for operators to provide recommendations on how to increase the organization's security posture to prevent further attacks by updating organizational policies and procedures.

Analytics will likely have false positives. False positives are going to occur when threat hunt teams are hunting for the malicious use of legitimate behaviors. Having knowledge of the network and normal baselines, outlined in outlined in Step 6: Testing Analytics, can help inform the analysis.

In addition to the techniques listed in Table 12, an analyst can constrain the analytic space by the time and terrain dimension. Operators can use the time dimension to analyze a shorter duration. At first, operators may look at a 24-hour period of activity and then gradually expand the time duration to a couple days, to a week, and then over a 30-day period. Performing this method could give the operator a different perspective of the activity that is occurring. As mentioned in Step 4: Understanding, Time, Behavior, and Cyber Terrain, operators may not have 30 days of data to review. The security sensor will only have data from the date of deployment, and operators may want to explore ingesting historical data if the network being investigated has been collecting logs.

Constraining analysis by the terrain can also be productive for the operator. In large networks with thousands of systems, its impractical for operators to attempt to look at all data at one time. Threat hunt teams can use the Crown Jewels Analysis to scope the systems that need a priority for analysis. The terrain dimension can also be used to assign a portion of the network to different operators as a focal point for their work.

**FALSE POSITIVES ARE
GOING TO OCCUR WHEN
THREAT HUNT TEAMS
ARE HUNTING FOR THE
MALICIOUS USE OF
LEGITIMATE BEHAVIORS.**

Occasionally, operators may be in a position when the analytic fails to return a useful result. This does not necessarily mean the analytic is flawed or that there is no malicious activity. It's possible the analytic is over-tuned and is too specific. Depending on the situation, operators could relax one of three dimensions to open the analytic to a wider set of results. The time frame could be extended with the analytic, expand the terrain to search over a larger portion of the network, or make slight changes to the behavior to see more results.

Evaluating Events

Once the number of events (or “hits”) generated by a given analytic is reduced to a number small enough, operators can devote some of their resources to pursue an investigation. This is a good point in the hunting process to incorporate the internal investigation methodologies that may have developed during the engagement. Evaluating hits can also be described as performing a triage of events. Triage is the process of performing preliminary analysis of an event to determine if an event is benign, suspicious, or malicious in order to assign degrees of urgency for operators to investigate. Operators would want to begin investigating malicious events before suspicious events. Benign events are known to be good activity and are used to further tune analytics to minimize the amount false positives. Suspicious events are neither known to be good nor bad and require further investigation. Malicious events are known to be bad and should be documented and reported.

When suspicious activity has been identified, widening the scope of the investigation (across time or the number of devices) to generate a broader set can help provide the context needed to determine if the activity is malicious. For example, a suspicious activity on one machine might actually be benign if the same activity occurs on all the machines in the network and has occurred for a considerable amount of time. To continue with the scheduled tasks example, if every host in the network is running the same scheduled tasks at startup, then it likely to be normal activity. Operators should consult with the system administrators to validate this assumption.

Contextual information is often needed to determine if an event is malicious or not. Adversaries do not perform actions in isolation, and thus the traces of activity they leave behind do not exist in isolation either. There will be a chain of causality to follow that can be used to connect seemingly disparate

events. Use the malicious activity model to provide insight into possible techniques that may occur as part of the chain of activity. Therefore, if a direct connection can be identified between the event under investigation and another event or piece of intelligence that is known to be malicious, the certainty that this event is also malicious increases significantly. Operators can look for the events that led to the scheduled tasks or events that are occurring because of the scheduled task. Some scheduled tasks used to maintain persistence and run at start up or at login. Hunt teams can look for associated processes that are created as a result of the scheduled tasks and look for internet connections that start at the same time.

Documenting Malicious Events

If the detected event is determined to be malicious, then it should be documented in such a way that the information can be shared between members of the teams as well other parties interested in the outcomes of the investigation. There are numerous ways that this information can be captured, here are some examples.

Adversary Timeline: The Adversary Timeline is a list of observed activity in chronological order. The list should contain the event that was observed and contextual information, such as users (if any) and host/IP address responsible. By adding this additional information, analysts can gain a greater appreciation of how the events are related. Once enough events have been identified, the team should consider trying to group the raw events into segments of activity. This will help the team gain context for the activity that may aid in understanding the overall adversary campaign. Adversary timelines are usually living documents that will continue to grow as new events are discovered during the investigation.

Host List: A list that contains relevant information regarding the various hosts that have been identified as being related to confirmed malicious activity. Some of the information that a team would want to capture includes hostnames, users, owners, IP addresses, and a reason for including the host.

User List: A list that contains information on users that have been confirmed as performing malicious activity. Additionally, consider adding users whose credentials may have been compromised, even if those credentials have not been tied to malicious activity. This may also include relevant information about the user that the hunt team may find useful like: Contact Information, Supervisor, Location, Role, Assigned Machines.

Malware List: A condensed list of the malware that may have been found within the environment. Any utilities or built-in programs that are being used by the adversary can also be tracked here. Here is a list of some of the information that should be captured: malware/program name, any aliases, general description, and other pertinent details about the malware.

Activity Graph: A map that describes the chain of activity between the various hosts identified. The purpose is to provide a visual representation of the malicious activity occurring on the network. The important pieces of information to capture on the graph are detailed below:

- Hosts that have been confirmed as having malicious activity take place on them. For this purpose, the hostname (rather than the IP address) is more useful as a given computer could have multiple IP addresses assigned to it for many reasons. However, there will likely be instances where an IP address is all that is available to use (e.g., an external C2 server).
- Network connections made between each of the hosts to show where the adversary pivoted in their operation. Capturing every network connection made between each host is unrealistic, so only a select few should be rendered. Initial malicious connections between two hosts are important to note, as this information helps establish how the adversary is moving around the network. As part of the connection information, it is important to capture the time/date of the connection as well as the protocol or method used.
- User credentials that were used (if any) are important to note as well. If legitimate user credentials are being used, then noting that can help inform directions that the hunt team needs to investigate further in. For example, if a user is observed making a malicious RDP connection to a host, but no information regarding what that user did on that host has been found yet, investigate it. Conversely, if a user's credentials are being used maliciously to navigate the network, then the hunt team needs to trace back those connections to try and find the moment where the credentials were compromised.

Gathering Contextual Information

Contextual information can be extremely important and for that reason collecting it is of utmost importance. Not only does it aid in understanding events that have been identified as malicious, but it can be used to drive direction for further investigation. Often, the most valuable information is that which can help to establish a chain of causality: what caused the event in question and what did the event cause in turn? By capturing these pieces of information, the team can focus their efforts on events that are directly tied to a known bad event. Events that precede a known malicious event should be considered very suspicious and events that were caused by it should be considered malicious. The following paragraphs, while by no means exhaustive, highlight things that an analyst should capture in relation to a given event. They provide a starting point for developing the team's own methods of connecting known malicious events to understand what happened.

OFTEN, THE MOST VALUABLE INFORMATION IS THAT WHICH CAN HELP TO ESTABLISH A CHAIN OF CAUSALITY: WHAT CAUSED THE EVENT IN QUESTION AND WHAT DID THE EVENT CAUSE IN TURN?

Related Processes

Identifying related processes can be invaluable. Through these relationships, it is relatively easy to establish chains of activity. The most important pieces of information to capture in this regard are “child” and “parent” process name/image paths, process IDs, and command lines. Additionally, the full command line of processes should be captured, if possible, as it often contains important information about the event. The arguments contained within Figure 37 on page 51 show how exactly that executable is being used and may also reveal additional information (e.g., any files that may have been used/modified or network connections that should be investigated further).

Network Information

Any network-related information that can be tied to a given event is also very important to capture as it will potentially reveal whether the event is part of a broader campaign and how it fits into the bigger picture. Without this context, an analyst is left with isolated series of activity with no direct ties to events happening on other hosts. The primary pieces of information that an analyst needs to capture relating to network activity are any IP addresses, ports, and any details regarding the content of the communication itself.

The last item in that list is difficult to define as it may vary considerably based on protocol and available information. For example, if the analyst observes a Secure Copy Protocol process create command to a remote address, then the analyst will have information regarding the file being transmitted. If, however, the analyst's visibility is limited to just netflow events, then the nature of the file being transmitted may be impossible to discern. Resolving any IP addresses identified to hostnames will also be beneficial for further investigations as well as coordinating with other team members.

System Files

Even in “file-less” attacks, adversaries will almost certainly interact with files on a system at some level. For example, adversaries may exfiltrate a user's documents or run an executable that, while an appropriate process for a typical Windows operation, is being run from an unusual directory. As an investigation progresses, it is important to keep track of pieces of information that are tied to relevant files. Ideally, these would be captured in a standardized data model, however some items that can be tracked are the file name, the file path of the executable, a hash of the file (especially if it is a binary or executable file), and any timestamp information. Some pertinent types of files include email attachments preceding other observed activity, creations/ deletions/modifications of files around the time of other events, and any files that are directly observed as being part of an event itself (e.g., any found within the command line of a malicious process start or observed being transmitted over a network connection).

User Information

User information can provide additional context regarding the adversary's activity. Not only can it reveal related information from the same data source, but it can be used to pivot across many of the host-based objects found in the data model. It can be used to identify additional processes being run by the same user, to look for files that that user was responsible for editing, as well as establish boundaries of activity by looking at log-in and log-out times and seeing how those log ins were accomplished. Other compromised hosts can also be identified by looking for the same activity. If the activity appears to be the same on both hosts, further investigation is likely warranted.

Investigating Malicious Events

To pursue a malicious hit, operators should investigate both backwards and forwards to find the activity which caused the hit (ideally back to the initial infection), as well as subsequent activity to determine the scope and scale of the adversary's actions.

In most cases, to begin pursuing the adversary, operators should work backwards to find the causes of the detected event. This will help determine the full scope of the activity, attribute the events to a specific adversary group, and gain the most useful knowledge for planning decisive response action. Ideally, operators will have the required data collection and analytic capability to determine each link in the causal chain of events leading to this initially detected event.

For example, on a Windows operating system, the responsible process could be found through identifying the parent process, schtasks command that scheduled this process start, the user event that triggered process start, or other methods as enumerated in ATT&CK's Execution Tactic. To trace the chain of causal execution across network traffic, the analyst might look for lateral movement methods like remote file copy, exploitation of remote services, or other methods.

If no causal events are found, the analyst will need to relax the requirement for finding evidence of each link in the causal chain. The analyst should consider the range of processes, systems, etc. that could have resulted in the event under consideration. For example, recent network connections, other activity by the same user or machine in the recent past, or other machines exhibiting identical behavior (e.g., same command line or network traffic).

In parallel with or after sufficient information has been obtained regarding causally preceding events, the hunt team should investigate caused or related subsequent activity. Similar to the investigation of preceding events, analysts should look first for evidence of directly caused activity such as child processes, file creations, or opened network connections. When needed, the analyst should expand the investigation to include other machines exhibiting identical behavior and other suspicious files, processes, or activity on the same system. As the investigation proceeds, analysts can consider the direct descendants of known-malicious activity to be malicious, while considering processes with a common parent as only suspicious pending further investigation and context.

IN MOST CASES, TO
BEGIN PURSUING
THE ADVERSARY,
OPERATORS SHOULD
WORK BACKWARDS TO
FIND THE CAUSES OF
THE DETECTED EVENT.

Throughout these pursuit investigations, analysts should continually refine the characterization of findings. As they gather more information, they should update a common knowledge repository (e.g., textual reporting, graph of activity) about the currently known chain of events outlined in Step 7: Documenting Malicious Events, to include information regarding whether they are indicative of a specific set of adversaries, whether this activity is indicative of a certain stage in the Cyber Attack Lifecycle and adversary intention. As new information is added to a shared repository, the team should also regularly determine what gaps in knowledge and/or visibility should be filled next and who and/or what could help fill them.

Concluding the Cyber Hunt Plan

At this point in the threat hunt process, the Cyber Hunt Plan has been developed and the analytics identified malicious activity. Threat hunt teams need to take the information from the Cyber Hunt Plan and the identified events and generate a report. Threat hunt teams can develop their own format for the report, but the report should have the elements that were defined in the plan. It is recommended the report have the following elements:

- Activity model with the techniques that were identified
- Hypothesis that was proven or disproven
- Data sources that were used to potentially identify the behavior
- Examples of events that identified the malicious behaviors
- Adversary timeline
- List of hosts, users, and/or processes that were compromised
- List of malware that may have been identified and the hosts associated with the malware
- If possible, the initial compromise vector
- Potential recommendation on how to defend against the identified threat

This report would be delivered to the next escalation tier in responding to confirmed security incidents. Some organizations have digital forensic teams to further identify malicious behavior on endpoints, or the report would escalate to the incident response team. Organizations should define processes with security incident handling to create an efficient and repeatable process to properly handle these incidents.

An opportunity presents itself for security teams to share information with a COI as the final report and any follow investigation uncovers artifacts. Being able to safely share information that will hold no attribution back to the internal network can empower the cyber community and potentially enhance any intelligence being used. This information may be unknown to the community. This will allow other security teams to potentially identify any malicious behaviors in their own network and may allow those teams to uncover new artifacts to be shared back into the community. Table 13 on the following page is a sample report that would be escalated based on the scheduled task example used throughout this manual.

TABLE 13. CYBER HUNT REPORT EXAMPLE

Cyber Hunt Plan	
Malicious Activity Model	T1053.005 Scheduled Task/Job: Scheduled Task—APT29 used named and hijacked scheduled tasks to establish persistence.
Summary	<p>The original hypothesis was that the adversary has used scheduled tasks to establish persistence on the network. An event was identified on September 30, 2021, at 14:44:05 that workstation “Test_Host” that an anomalous task was created. Reviewing the baseline on approved scheduled tasks shows this is a potentially malicious task as it does not conform to naming conventions and not listed on any baseline. The task was created under the user “User1” potentially indicating that the account has been compromised. No other scheduled tasks events were identified on the host. Running the hashes through OSINT tools shows the schtasks.exe as the legitimate tool. There were no other events observed to collect information on the executable that was scheduled.</p> <p>This event indicates that a malicious actor has achieved persistence to maintain access to the system across restarts, credential changes, or other interruptions. No malware was identified from the event logs, but it is recommended to perform a forensic investigation to verify the content of the executable seen in the scheduled task. It is recommended the account “User1” is blocked on the network and the host until further investigation.</p>
Artifacts	<p>Hosts: Test_Host Users: User1 CommandLine: C:\Windows\system32\schtasks.exe” /create /tn Evil_Schedule /sc onlogon /tr “cmd.exe /c calc.exe hashes: MD5=796B784E98008854C27F4B18D287BA30 SHA256=356280CCA63CA5E887FDBE5CB4105A53341FBAC9219EFC51621DF9BA8EE9838B IMPHASH=ECCE05491F2E8F279F4790BCB1318C05 Image: C:\Windows\System32\schtasks.exe</p>
Data Source Used	Sysmon Event ID 1 Process Create
Analytics Used	process.name:schtasks.exe and process.args.(“/create” or “-create” or “/S” or “-s” or “/run” or “/change” or “-change”
Events	<pre>Message : { Process Create } RuleName : UtcTime : 2021-09-30 14:44:05.032 ProcessId : (XXXXXXXX-6855-6155-5722-000000000500) ProcessInfo : { Image: C:\Windows\System32\schtasks.exe FileVersion: 10.0.19041.900 (WinBuild.190818.0800) Description: Task Scheduler Configuration Tool Product: Microsoft Windows Operating System Company: Microsoft Corporation OriginalFileName: schtasks.exe CommandLine: "C:\Windows\System32\schtasks.exe" /create /tn Evil_Schedule /sc onlogon /tr "cmd.exe /c calc.exe" CurrentDirectory: C:\Users\User1\Desktop\ User: Test_Host\User1 LogonId: (XXXXXXXX-0770-6155-CFE7-070000000000) TerminationPolicy: 4 IntegrityLevel: High MimeType: MS-PowerShell ParentProcessId: (XXXXXXXX-0C13-6155-4C22-000000000500) ParentProcessInfo: { Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe ParentCommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" } } Id : 1 Version : 0 Qualifiers : 0 Level : 4 Task : 3 DocObj : 0 Keywords : ["9223372036854775808"] RecordId : 100 ProviderName : Microsoft-Windows-Sysmon ProviderId : {"5780360F-C22A-430B-B94C-A9FD8808DD40"} LogName : Microsoft-Windows-Sysmon/Operational ProcessId : 1744 ThreadId : 17968 MachineName : Test_Host UserId : S-1-5-18 TimeCreated : 9/30/2021 9:44:05 AM ActivityId : RelatedActivityId : Containing : { Microsoft-Windows-Sysmon/Operational } MatchedQueryIds : [1] Bookmark : System.Diagnostics.Eventing.Reader.EventBookmark LevelDisplayName : Information SourceDisplayName : Sysmon TaskDisplayName : Process Create (rule: ProcessCreate) OperationDisplayName : Properties : { System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty, System.Diagnostics.Eventing.Reader.EventProperty, ... }</pre>

Responding to the Security Incident

Throughout this process, threat hunt teams must be mindful of the courses of actions possible for responding to the intrusion under consideration by the network owners and tailor the investigation accordingly. There may be different choices made depending on whether the intent is to determine the full scale and scope of the intrusion versus quickly attributing the activity to an adversary group. As a result, operators may alternately prioritize finding the source of the activity, finding the subsequently targeted systems, or performing deep forensic analysis to better understand the characteristics of the activity or artifacts likely to aid in attribution.

Over time, the knowledge gained by the hunt will be sufficient to make decisions on courses of action (e.g., quarantine, movement of the adversary to a deception environment, placement of honey credentials or misinformation, or perimeter blocking). This may occur when the full extent of the adversarial activity is known, or when the defensive team's knowledge and ability to effectively defend and respond can render the adversary's attack ineffective. Operators must strike the right balance between waiting too long to act and acting prematurely. Too much emphasis on learning the full extent of the activity may hamper timely responsive action. Acting before sufficient knowledge is gained could result in tipping one's hand to the adversary without having significant impact on their presence in the network or their ability to accomplish their objectives. If operators don't identify the full intent of the malicious activity, the adversary could regain access through a vulnerability that wasn't patched, or they may have secondary access. This is a strategic decision that should incorporate an understanding of the adversary's activity, their intent and capabilities, and the potential or actual impact to the defended environments.

Operators should develop response playbooks to provide guidance for action when an incident occurs. Having response playbooks already developed also provides tasks that can be practiced prior to hunting. The list below provides resources for incident response playbooks:

- **Atomic Threat Coverage RE&CT Project:** <https://atc-project.github.io/atc-react/>
- **Incident Response Consortium:** <https://www.incidentresponse.com/playbooks/>
- **GuardSight, Inc. Playbooks Battle Cards:** https://github.com/guardsight/gsvsoc_cirt-playbook-battle-cards
- **ElysiumSecurity Playbooks:** <https://github.com/elysiumsecurity/ltd/IRM>

If malicious activity has been identified in the environment, Incident Response may need to be performed to remove the malicious activity from the environment. There are 6 steps for Incident Response process: Preparation, Identification, Containment, Eradication, Recovery, and Lessons Learned. In some situations, the operator may be the one that identifies an incident for the Incident Response team. This is where documentation becomes vital for scoping out the incident. The who, what, when, where, and why need to be documented in order to learn from the incident. With threat hunting falling within the Detection and Analysis section of the NIST Incident Response, having this information will assist with other parts of the Incident Response Lifecycle. Once the Incident Response teams starts the Containment, Eradication, and Recover phase, the operator is able to assist and validate by monitoring host and network behavior to make sure the malicious activity has been removed.

Incident Response is an organized and repeatable approach to addressing a confirmed breach or cyberattack. Organizations should define incident response plans, policies, and procedures to create a repeatable process to address any confirmed detections through DCO and threat hunt engagements. Reviewing NIST SP 800-61 rev. 2 Computer Security Incident Handling Guide for developing an incident response program will aid in developing an incident response program (NIST, 2012) The publication can be found at <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>

Assess Analytics and Hunt Process

After executing an analytic for a hunt, the analytic needs to be assessed on how well it worked, its impact on the success of the hunt, and what can be improved. When assessing analytics, operators want to provide some measurement to show success and how the analytics align to the goals and objectives for the hunt.

Some of the metrics that can be assessed include the list below:

- Time spent (hours)
- Number of incidents identified
- Number of analytics updated
- Number of security recommendations provided to the network owner
- Number of vulnerabilities discovered
- Defining the scope (This can be used to identify number of hosts, number servers, number of users, or how much of the network was analyzed.)

Additional Considerations

The 7-Step process as described in the previous sections explains the process from beginning to end. Once operators have the analytics developed for an identified environment and translated for the specific SIEM, operators don't have to go back through the process every time for a threat hunt engagement. Operators only have to start from the beginning of the process when the malicious activity model changes because new intelligence has been identified or the environment changes and new analytics need to be developed to adapt to the changes.

Another source that is becoming popular with threat detection are Sigma analytics. Sigma analytics are generic rules that can be converted and shared to run against different data analytics tools. Sigma analytics are written in YAML. Sigma addresses the interoperability issues with conflicting query languages in different data analytic tools preventing vendor lock-in. Sigma can be leveraged with MISP through the functionality modules Sigma2MISP to import Sigma rules to MISP events and Sigma Importer—Sigma to convert specific data sources into the Sigma format. There are Sigma analytics already developed and being shared from SigmaHQ and OTRF Threat Hunter Playbook. Operators can review analytics for a specific TTP they're looking to detect to potentially identify additional data sources not listed in other sources and specific artifacts that may provide detection capabilities within their environment.

Threat hunt teams should develop and maintain a use case library that includes the analytics developed and includes additional information to help support the analytics. A Use Case is a documented approach to a situation in which an analytic or additional form of detection can potentially be used. A useful example for formatting a Use Case is the Alerting Detection Strategies (ADS) Framework used by the Palantir Incident Response Team. The ADS Framework provides a natural language template which helps frame hypothesis generation, testing, and management. Table 14 outlines the sections of the structure of a Use Case (Palantir, n.d.). The ADS framework can be found at <https://github.com/palantir/alerting-detection-strategy-framework>.

SIGMA ADDRESSES
THE INTEROPERABILITY
ISSUES WITH
CONFLICTING QUERY
LANGUAGES IN
DIFFERENT DATA
ANALYTIC TOOLS
PREVENTING VENDOR
LOCK-IN.

TABLE 14. USE CASE FRAMEWORK

Section	Definition
Goal	The goal is the intended purpose of the Use Case. It is a simple, plaintext description of the type of behavior attempting to detect in the Use Case.
Categorization	The categorization is a mapping of the Use Case to the relevant entry in ATT&CK. ATT&CK provides a language for various post-exploitation techniques and strategies that adversaries might use. Mapping to the ATT&CK framework allows for further investigation into the technique, provides a reference to the areas of the kill chain where the ADS will be used and can further drive insight and metrics into alerting gaps. In the environment, there is a knowledge base which maps all of the ADS to individual components of the MITRE ATT&CK framework. When generating a hypothesis for a new alert, an engineer can simply review where the network is strongest—or weakest—according to individual ATT&CK techniques. When selecting a MITRE ATT&CK category, please select both the parent and child category (e.g., Credential Access/Brute Force). Additional categorizations can include specific adversary groups (APT29, Fancy Bear, etc.) and the malware that the Use Case was designed for
Strategy Abstract	The strategy abstract is a high-level walkthrough of how the Use Case functions. This describes what the Use Case is looking for, what technical data sources are used, any enrichment that occurs, and any false positive minimization steps.
Technical Context	<p>Technical Context provides detailed information and background needed for a responder to understand all components of the Use Case. This should appropriately link to any platform or tooling knowledge and should include information about the direct aspects of the alert. The goal of the Technical Context section is to provide a self-contained reference for a responder to make a judgement call on any potential detection, even if they do not have direct subject matter expertise on the Use Case itself.</p> <ul style="list-style-type: none"> ▪ Pseudocode analytic ▪ KQL version of analytics ▪ Data sources required ▪ Description of dashboard ▪ Filtering strategies
Blind Spots and Assumptions	Blind Spots and Assumptions are the recognized issues, assumptions, and areas where an analytic may not fire. No analytic is perfect and identifying assumptions, and blind spots can help other users understand how an analytic may fail to fire or be defeated by an adversary.
False Positives	<p>False Positives are the known instances of an analytic misfiring due to a misconfiguration, idiosyncrasy in the environment, or other non-malicious scenario. The False Positives section notes uniqueness to the environment and should include the defining characteristics of any activity that could generate a false positive alert. These false positive hits should be suppressed within the SIEM to prevent future hits when a known false positive event occurs. Each analytic needs to be tested and refined to remove as many false positives as possible before it is put into production. False positive minimization relies on looking at several principles of the strategy and adjusting, such as:</p> <ul style="list-style-type: none"> ▪ Add an additional component to the rule to maximize true positives ▪ Remove common false positives through pattern ▪ Back-end filtering to store indices of expected false positives <p>Ideally, operators want to have the fewest false positives possible while maintaining the spirit of the rule. If a low false positive rate cannot be reached, the analytic may need to be broken down, refactored, or entirely discarded</p>

Validation	<p>Validation are the steps required to generate a representative true positive event which triggers this analytic. This is similar to a unit test and describes how an engineer can cause the analytic to fire. This can be a walkthrough of steps used to generate an alert, a script to trigger the analytic (such as Red Canary's Atomic Red Team Tests), a scenario used in an alert testing and orchestration platform, or by using MITRE's Caldera. Each analytic must have true positive validation. This is a testing process designed to prove the true positives are detected. True positive validation relies on generating a scenario in which the detection strategy is testing, and then validating in the tool. To perform positive validation, complete the steps:</p> <ol style="list-style-type: none"> 1. Generate a scenario where a true positive would be generated. 2. Document the process of the testing scenario. 3. From a testing device, generate a true positive analytic hit. 4. Validate the true positive analytic hit was detected by the strategy. 5. If operators are unable to generate a true positive analytic hit, the alert may need to be broken down, refactored, or entirely discarded.
Priority	<p>Priority describes the various alerting levels that a Use Case may be tagged with. While the analytic itself should reflect the priority when it triggers on events through configuration in the SIEM (e.g. High, Medium, Low), this section details the criteria for the specific priorities. Having prioritizations can help with organizing hunts and making decisions on which analytics on the most important to hunt for.</p>
Response	<p>These are the general response steps in the event that this analytic gets a hit. These steps instruct the next responder on the process of triaging and investigating an alert. Include the additional contextual information that should be collected when the analytic gets a hit.</p>
Additional Resources	<p>Additional Resources are any other internal, external, or technical references that may be useful for understanding the Use Case.</p>

Threat hunt teams should have a process for the development of these Use Cases to include the onboarding of Use Cases from external organizations. Including an onboarding process for external Use Cases is necessary to ensure the external information is put into the organizational format and includes the required additional information. The Use Cases can include additional forms of detection such as Suricata signatures, Yara rules, and Sigma rules.

The following is an example of a Use Case Development Cycle that can be used in parallel to the 7-Step process already described in this document.

USE CASE DEVELOPMENT CYCLE

1. **Requirements:** The step is informed by Threat Intelligence or externally sourced detections, analytics, and Use Cases. This step also takes the feedback from the cycle to establish requirements to update or develop Use Cases.
2. **Identify Data Sources:** This step is used to identify the data sources required.
3. **Design Logic:** This step is where the pseudocode analytic, KQL, Suricata signature, Yara rule, or Sigma rule is developed.
4. **Test and Validate Logic:** This step is when the logic developed for the Use Case is tested and validated. Operators want to use a resource that will enable threat emulation, such as Atomic Red Team or Caldera.
5. **Proof of Concept:** This step is used to develop a proof of concept that be turned into a Use Case.
6. **Use Case Design:** This step is used to take the proof of concept and put into the organization's Use Case template.
7. **Train Analysts:** The step is used to train the analysts, get their acceptance, and make the Use Case ready to go into production.
8. **Promote Into Production:** This step is when the threat hunt team promote the Use Case into production, and analysts start operationalizing the Use Case.
9. **Finetune:** The is step is while the Use Case is being used operationally, and small updates are made to improve performance.
10. **Periodic Review and Feedback:** This step is used to review Use Cases that have been in production for an extended period of time, such as one year. The step also includes the feedback from the analysts on the how well the Use Case works, if there are any issues, or changes that should be made. The feedback is used to establish requirements for the Requirements step.

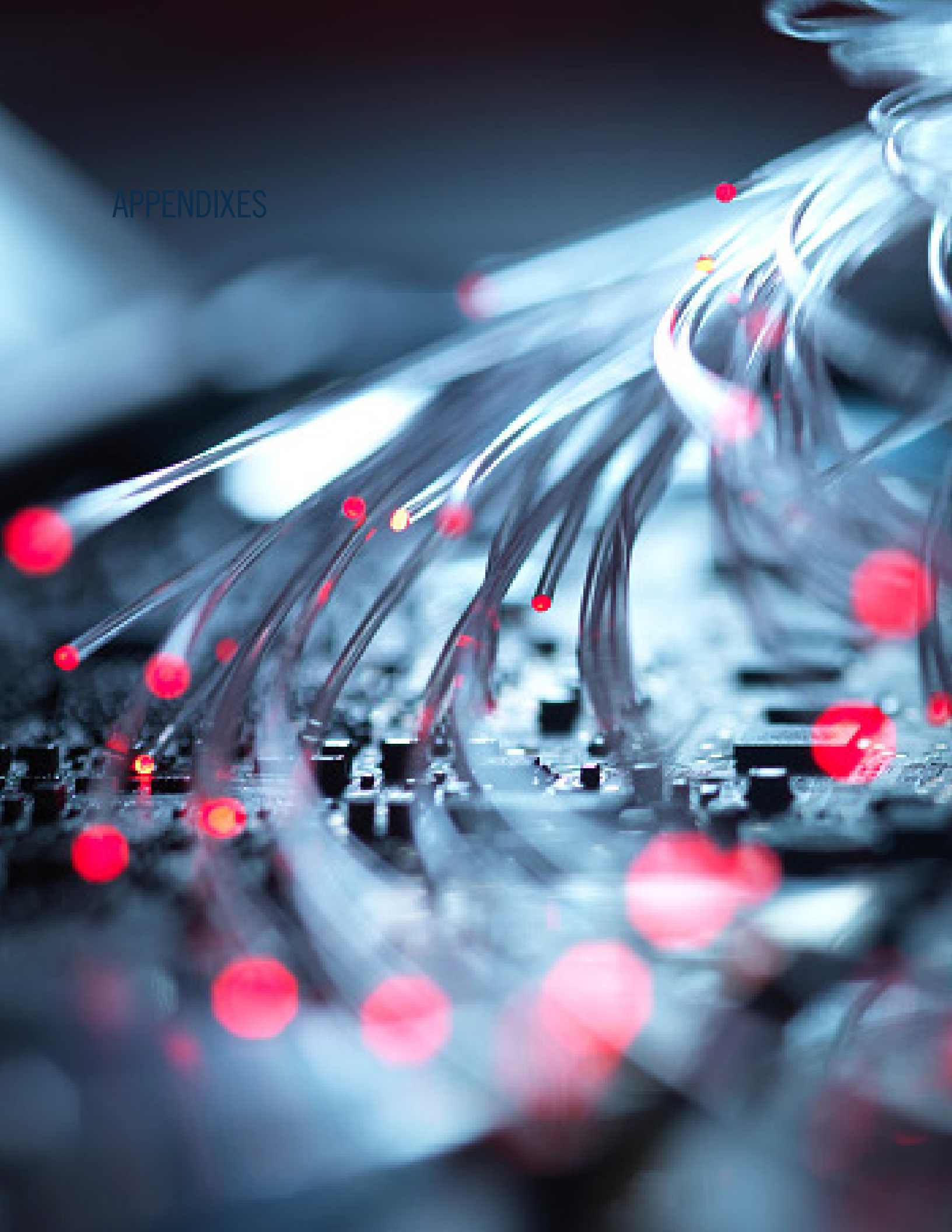
Using a resource such as a Git repository, teams can manage a development cycle for the organization's Use Cases. A Git repository also provides version control and enables the sharing of Use Case with other organizations.

The goal of these additional considerations to provide additional tools that will aid in maturing threat hunting operations. Being able to measure current operations and apply that information to a defined maturity model can aid organizations in developing effective roadmaps to maturity into their threat hunting operations. David Bianco's Hunt Maturity Model (HMM) considers the quantity and quality of data collected, ability to visualize and analyze various types of data, and different types of automated analytics operators can apply to enrich analytic insights. The HMM describes five levels of organizational hunting capability (Sqrll, 2015):

- **HMO—Initial:** Organization relies primarily on automated alerting through IDS, IPS, SIEM, or antivirus to detect malicious activity across the enterprise. Organizations may incorporate threat intelligence feeds or create internal signatures or indicators but primarily focused on alert resolution.
- **HM1—Minimal:** Organization still relies primarily on automated alerting to drive incident response procedures but are performing some routine collection of actionable data. The goal is to work towards intel-driven detection through extracting artifacts from intelligence reports and incorporating that data into a centralized logging solution for threat detection.
- **HM2—Procedural:** Organizations are able learn and apply procedures developed by other organizations or the community and make necessary changes to tailor those procedures to the environment under investigation. However, the organization is not capable of creating new procedures.
- **HM3—Innovative:** Organizations have multiple threat hunters with expertise in different domains and can apply those techniques to identify malicious activity. Instead of relying on procedures developed from other organizations or the community, the internal team create and publish the procedures.
- **HM4—Leading:** Organizations can apply the skillsets of HM3 teams into automated content. Successful threat hunting campaigns are operationalized and automated to potentially discover malicious activity quickly.

BEING ABLE TO
MEASURE CURRENT
OPERATIONS AND APPLY
THAT INFORMATION TO
A DEFINED MATURITY
MODEL CAN AID
ORGANIZATIONS IN
DEVELOPING EFFECTIVE
ROADMAPS TO
MATURITY INTO THEIR
THREAT HUNTING
OPERATIONS

APPENDIXES



Appendix A: APT28 ATT&CK Techniques

TABLE A-1. APT ATT&CK TECHNIQUES

MITRE ATT&CK Matrix Tactics	MITRE ATT&CK Techniques	MITRE ATT&CK Sub-Techniques	MITRE ATT&CK Number	Data Sensor Type	ADCS Tools
Reconnaissance	Active Scanning	Vulnerability Scanning	T1595.002	Network-based and Host-based data	Sysmon Zeek Suricata
	Gathering Victim Identity Information	Credentials	T1589.001	Network-based and Host-based data	Sysmon Zeek Suricata
	Phishing for Information	N/A	T1598	None	OSINT
Resource Development	Phishing for Information	N/A	T1598	None	OSINT
Initial Access	Exploiting Public Facing Application	N/A	T1190	Network-based and Host-based data	Sysmon Zeek Suricata
	Phishing	Spear-phishing Attachment	T1566.001	Network-based and Host-based data	Sysmon Zeek Suricata
		Spear-phishing Link	T1566.002	Network-based and Host-based data	Sysmon Zeek Suricata
	Replication Through Removable Media	N/A	T1091	Host-based data	Sysmon
	Trusted Relationship	N/A	T1199	Host-based data	Sysmon
	Valid Accounts	Domain Accounts	T1078.002	Host-based data	Sysmon
	Command and Scripting Interpreter	PowerShell	T1059.001	Host data	Sysmon
Execution	Command and Scripting Interpreter	Windows Command Shell	T1059.003	Host-based data	Sysmon
		N/A	T1203	Host-based data	Sysmon
	Exploitation for Client Execution	N/A	T1203	Host-based data	Sysmon
	Inter-Process Communication	Dynamic Data Exchange	T10559.002	Host-based data	Sysmon
	User Execution	Malicious File	T1204.002	Host-based data	Sysmon
		Malicious Link	T1204.001	Network-based and Host-based data	Sysmon Zeek Suricata
Persistence	Boot or Logon Autostart Execution	Registry Run Keys /Startup Folder	T1547.001	Host-based data	Sysmon
	Boot or Logon Initialization Scripts	Logon Scripts (Windows)	T11037.001	Host-based data	Sysmon
	Event Triggered Execution	Component Object Model Hijacking	T1546.015	Host-based data	Sysmon
	Office Application Startup	Office Test	T1137.002	Network-based and host-based data	Sysmon
	Pre-OS Boot	Bootkit	T1542.003	Host-based data	Sysmon
	Valid Accounts	N/A	T1078	Host-based data	Sysmon

Privilege Escalation	Access Token Manipulation	Token Impersonation/Theft	T15134.001	Host-based data	Sysmon
	Boot or Logon Autostart Execution	Registry Run Keys/Startup Folder	T1547.001	Host-based data	Sysmon
	Boot or Logon Initialization Scripts	Logon Script (Windows)	T14037.001	Host-based data	Sysmon
	Event Triggered Execution	Component Object Model Hijacking	T1546.015	Host-based data	Sysmon
	Exploitation for Privilege Escalation	N/A	T1068	Host-based data	Sysmon
	Valid Accounts	N/A	T1078	Host-based data	Sysmon
Defense Evasion	Access Token Manipulation	Token Impersonation/	T1134.001	Host-based data	Sysmon
	Deobfuscate/Decode Files or Information	Theft	T1140	Host-based data	Sysmon
	Exploitation for Defense Evasion	N/A	T1211	Host-based data	Sysmon
	Hide Artifacts	Hidden Files and Directories	T1564.001	Host-based data	Sysmon
		Hidden Window	T1564.003	Host-based data	Sysmon
	Indicator Removal on Host	Clear Windows Event Logs	T1070.001	Host-based data	Sysmon
		File Deletion	T1070.004	Host-based data	Sysmon
		Timestomp	T1070.006	Host-based data	Sysmon
	Obfuscated Files or Information	N/A	T1027	Host-based data	Sysmon
	Pre-OS Boot	Bootkit	T1542.003	Host-based data	Sysmon
	Rootkit	N/A	T1014	Host-based data	Sysmon
	Signed Binary Proxy Execution	Rundll32	T1218.011	Host-based data	Sysmon
	Template Injection	N/A	T1221		Sysmon Zeek Suricata
Credential Access	User Alternate Authentication Material	Application Access Token	T1550.001	Host-based data	Sysmon
		Pass the Hash	T1550.001	Host-based data	Sysmon
	Valid Account	N/A	T1078	Host-based data	Sysmon
	Brute Force	Password Guessing	T1110.001	Host-based data	Sysmon
	Input Capture	Password Spraying	T1110.003	Host-based data	Sysmon
	Network Sniffing	N/A	T16056.001	Host-based data	Sysmon
	OS Credential Dumping		T1040	Host-based data	Sysmon
	Steal Application Access Token	N/A	T1003.001	Host-based data	Sysmon

Discovery	File and Directory Discovery	N/A	T1083	Host-based data	Sysmon
	Network Sniffing	N/A	T1040	Host-based data	Sysmon
	Peripheral Device Discovery	N/A	T1122	Host-based data	Sysmon
	Process Discovery	N/A	T1057	Host-based data	Sysmon
Lateral Movement	Exploitation of Remote Services	N/A	T1210	Network-based and Host-based data	Sysmon Zeek Suricata
	Replication Through Removable Media	N/A	T1091	Host-based data	Sysmon
	Use Alternate Authentication Material	Application Access Token	T1550.001	Host-based data	Sysmon
		Pass the Hash	T1550.002	Host-based data	Sysmon
Collection	Archive Collected Data	N/A	T1560	Host-based data	Sysmon
	Automated Collection	N/A	T1119	Host-based data	Sysmon
	Data from Information Repositories	Sharepoint	T1213.002	Host-based data	Sysmon
	Data from Local System	N/A	T1005	Host-based data	Sysmon
	Data from Removable Media	N/A	T1025	Host-based data	Sysmon
	Data Staged	Local Data Staging	T1074.001	Host-based data	Sysmon
	Email Collection	Remote Email Collection	T1114.002	Network-based and Host-based data	Sysmon Zeek Suricata
	Input Capture	Keylogging	T1056.001	Host-based data	Sysmon
	Screen Capture	N/A	T1113	Host-based data	Sysmon
	Command and Control	Application Layer Protocol	Mail Protocols	T1071.003	Network-based data
Web Protocols			T1071.001	Network-based data	Zeek Suricata
Communication Through Removable Media		N/A	T1092	Network-based data	Zeek Suricata
Data Obfuscation		Junk Data	T1001.001	Network-based data	Zeek Suricata
Encrypted Channel		Symmetric Cryptography	T1573.001	Network-based data	Zeek Suricata
Ingress Tool Transfer		N/A	T1105	Network-based data	Zeek Suricata
Proxy		External Proxy	T1090.002	Network-based data	Zeek Suricata
		Multi-hop Proxy	T1090.003	Network-based data	Zeek Suricata
Web Service		Multi-hop Proxy	T1102.002	Network-based data	Zeek Suricata
Exfiltration		Exfiltration Over Web Service	N/A	T10567	Network-based and Host-based data

Appendix B: APT28 & APT29 Compare Open Source

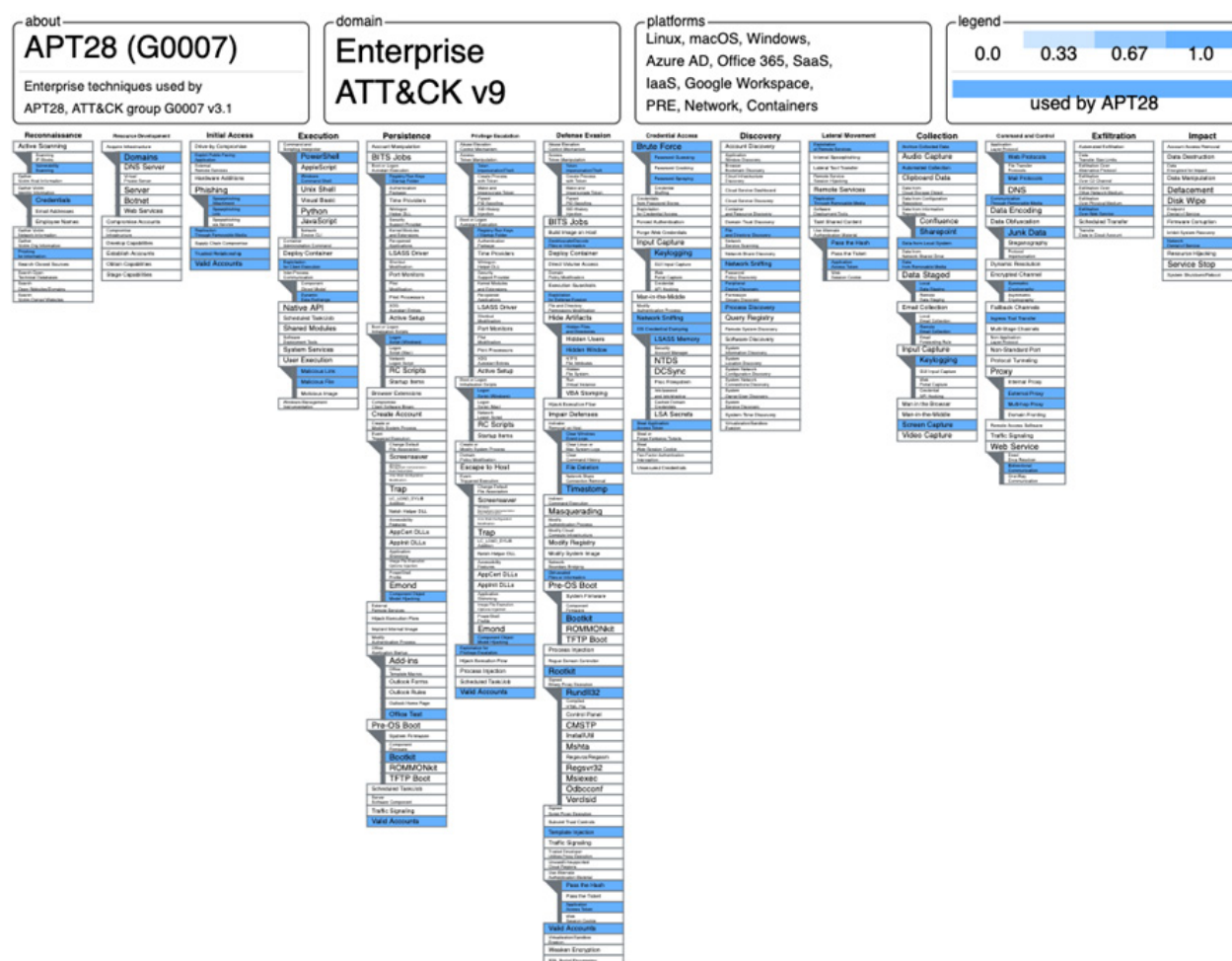


FIGURE B-1. APT28 ATT&CK MAPPING

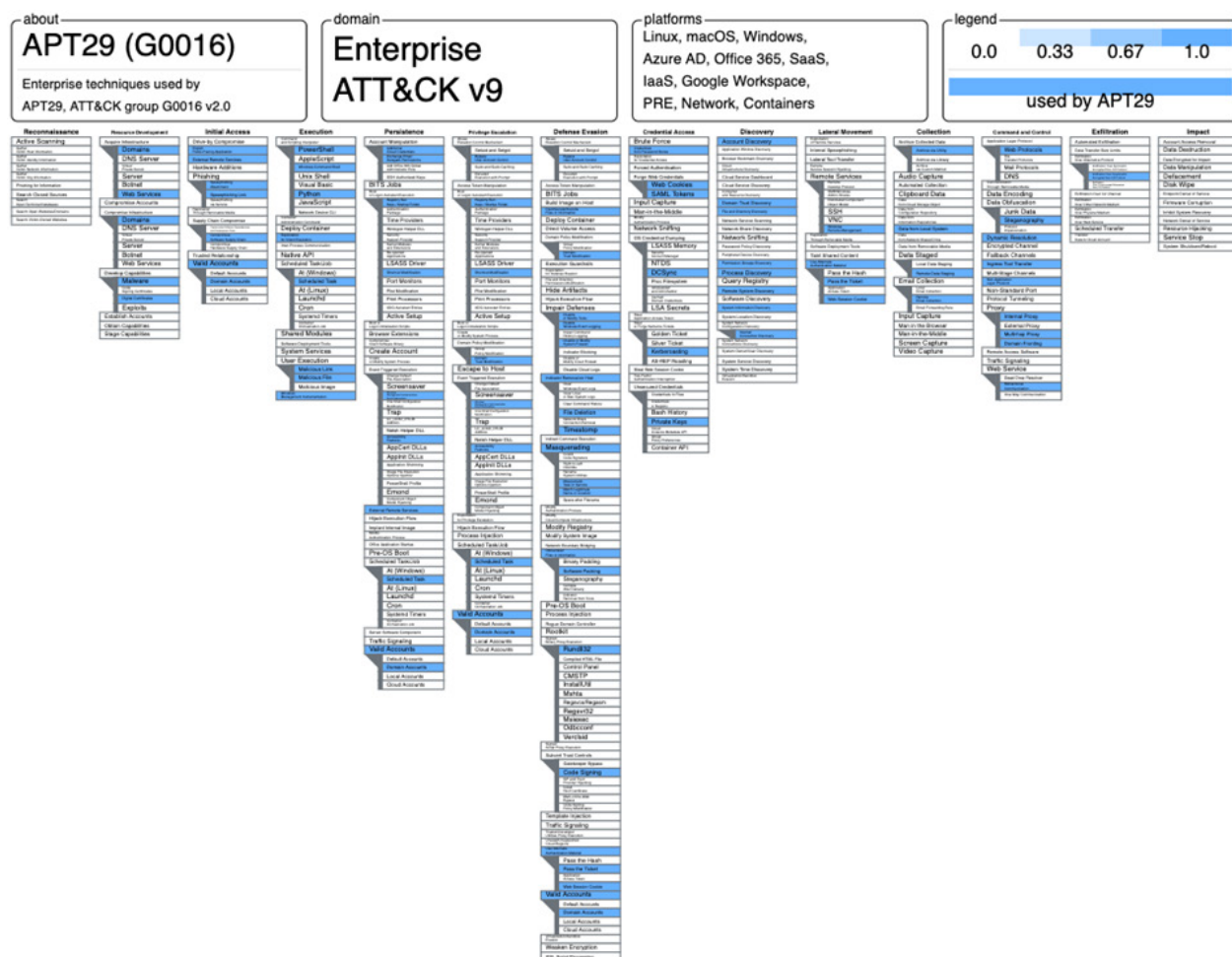


FIGURE B-2. APT29 ATT&CK MAPPING

Appendix C: Categorized Tools

MITRE derived the software categories from a Cyber Mission Technical Report that was developed for the Navy on their Deployable Mission Support Systems. The list of tools represents the capabilities that the MITRE Team collected during the initial survey, with no assessment of their value.

Discovery Tools Tools that are intended to gather data about a given network, including hosts on a typical network. Data may be passively and actively collected.

TABLE C-1. DISCOVERY TOOLS

Free and open-source	Zeek (Bro)	Grass Marlin (Archived)
	Nmap	
Commercial, with cost	Network Visualization Suite	SolarWinds Engineers Toolset
	Dark Ether	

Network-based Digital Forensics and Incident Response (DFIR) Tools that use passive and active methods to generate an evaluation of the defended critical assets and processes.

TABLE C-2. NETWORK-BASED DFIR

Free and open-source	Suricata	Wireshark
	Zeek (Bro)	Arkime (Moloch)
Commercial, with cost	ThreatPinch Lookup	
	SolarWinds Engineers Toolset	

Host-based DFIR Tools that use passive and active methods to generate an evaluation of the defended critical assets and processes.

TABLE C-3. HOST-BASED DFIR

Free and open-source	Google Rapid Response	Sysmon
	OSSEC Agents	Regshot
	SysInternals Suite	Bluespawn
	ThreatPinch Lookup	Osquery
	Signature Base	
Commercial, with cost	Endgame	Carbon Black
	FTK Registry Viewer	FireEye HX
	Surge Collect Pro	PE Explorer

Vulnerability Assessment Tools that analyze a given environment to identify weaknesses

TABLE C-4. VULNERABILITY ASSESSMENT TOOLS

Free and open-source	Nmap	OpenVAS
Commercial, with cost	Acunetix Vulnerability Scanner	Dark Ether
	Titania Nipper Studio 4000	Burp Suite Pro
	Tenable Nessus Professional	Rapid7 Nexpose

Mitigation, Clearing, and Remediation Tools that disrupt rogue processes, malware, adversary C2, and specified network connections on defended systems.

TABLE C-5. MITIGATION, CLEARING, AND REMEDIATION TOOLS

Free and open-source	Google Rapid Response	Bluespawn
Commercial, with cost	Carbon Black	Endgame
	FireEye HX	

Forensics and Malware Analysis Tools that gather and preserve evidence from a potential attack on a system as well as analyze suspicious code to understand its functions and potential impact.

TABLE C-6. FORENSICS AND MALWARE ANALYSIS TOOLS

Free and open-source	Autopsy	Cuckoo Sandbox
	FTK Imager	SIFT
	REMnux	ThreatPinch Lookup
Commercial, with cost	FTK Registry Viewer	PE Explorer
	EnCase	IDA Pro
	Surge Collect Pro	EnCase Endpoint Investigator

Continuous Monitoring Tools that conduct ongoing observation and analysis of the operational states of critical systems to provide situational awareness for response actions.

TABLE C-7. CONTINUOUS MONITORING TOOLS

Free and open-source	Suricata	ELK Stack
	Logstash	Kibana
	OSSEC Agents	Grafana
	Security Onion	Moloch
	Sysmon	Zeek (Bro)
	Regshot	RockNSM
	Bluespawn	Threat Ingestor
	CAVES	
Commercial, with cost	SolarWinds Engineers Toolset	Splunk
	Carbon Black	Splunk Server
	Splunk Universal Forwarder	Endgame
	FireEye HX	

Event Correlation and Analysis Tools that provide insight into the root cause(s), technical details, and potential impacts of a cyberspace incident.

TABLE C-8. EVENT CORRELATION & ANALYSIS TOOLS

Free and open-source	ELK Stack	Logstash
	Kibana	Grafana
	Security Onion	RockNSM
	CAVES	
	Fast Incident Response (FIR)	TheHive
Commercial, with cost	Splunk	Splunk Server
	Splunk Universal Forwarder	Elastic_Xpack
	HP ArcSight	

Event Correlation and Analysis Tools that provide insight into the root cause(s), technical details, and potential impacts of a cyberspace incident.

TABLE C-9. THREAT EMULATION TOOLS

Free and open-source	MITRE CALDERA
Commercial, with cost	Cobalt Strike

Cyber Threat Intelligence Definition: Tools that can generate or produce CTI that enables an operator to make informed decisions during analysis and operations.

TABLE C-10. CYBER THREAT INTELLIGENCE (CTI) TOOLS

Free and open-source	MISP	Threat Ingestor
Commercial, with cost		

Suite of Tools A collection of tools found in other categories, bundled together as one solution

TABLE C-11. SUITE OF TOOLS

Free and open-source	Suricata	Wireshark
	Zeek (Bro)	Arkime (Moloch)
	ThreatPinch Lookup	
Commercial, with cost	ThreatPinch Lookup	

Administrative Tools that help any operator at any point in the process of cyber hunts (i.e., chat services, documentation, software packaging).

TABLE C-12. ADMINISTRATIVE TOOLS

CAPEs	7zip	Atom
Putty	WinSCP	VMWare Workstation Pro
Microsoft Office Pro 2016	Microsoft Visio Studio Pro 2016	CentOS7
Kubernetes	Docker	Flannel
Xwiki	Mattermost	Redmine
Windows 10 (AMNET Image)	Windows 10 (Classroom Image)	Kali/Ubuntu
Yeti	Threathunter Playbook	Awesome Threat Detection
Awesome Yara	ATT&CK Datamap	Cisco ASA
Clonezilla	Sandstorm	

Free and Open-Source Capabilities

This section of the appendix describes the free and open-source capabilities, provides links to documentation for reference, and links to training if available.

▪ **Arkime (Moloch)**

Description

- Arkime (formerly Moloch) is a large scale, open-source, indexed packet capture and search tool.

Documentation

- Arkime documentation <https://arkime.com/learn>
- Training: Arkime YouTube channel https://www.youtube.com/channel/UCCtFDN7jSW_Np6iOZ_B6t8Q/videos

▪ **Autopsy**

Description

- Autopsy is the premier open-source forensics platform, which is fast, easy-to-use, and capable of analyzing all types of mobile devices and digital media. Its plug-in architecture enables extensibility from community-developed or custom-built modules. Autopsy evolves to meet the needs of hundreds of thousands of professionals in law enforcement, national security, litigation support, and corporate investigation.

Documentation

- Autopsy User Documentation <http://sleuthkit.org/autopsy/docs/user-docs/4.18.0/>

Training

- DFIR Science YouTube channel <https://www.youtube.com/c/DFIRScience/featured>

▪ **BluespawN**

Description

- BLUESPAWN is an active defense and endpoint detection and response tool, which means it can be used by defenders to quickly detect, identify, and eliminate malicious activity and malware across a network.

Documentation

- BluespawN github page <https://github.com/ION28/BLUESPAWN>

Training

- Defcon 28 Blue Team Village Presentation <https://github.com/ION28/BLUESPAWN/blob/master/docs/media/Defcon28-BlueTeamVillage-BLUESPAWN-Presentation.pdf>
- Blue Team Village YouTube <https://www.youtube.com/watch?v=mO4GrM8dapQ>

▪ **CALDERA**

Description

- CALDERA™ is a cybersecurity framework developed by MITRE that empowers cyber practitioners to save time, money, and energy through automated security assessments.

Documentation

- CALDERA Github <https://github.com/mitre/caldera>
- CALDERA Documentation <https://caldera.readthedocs.io/en/latest/>

Training

- MITRE CALDERA YouTube Playlist https://www.youtube.com/playlist?list=PLkTApXQou_8KFTzR7KqDJh-ndMO39PYnB

▪ CAPES

Description

- CAPES is an operational-focused service hub for segmented, self-hosted, and offline (if necessary) incident response, intelligence analysis, and hunt operations.
- Includes Rocketchat, Etherpad, Gitea, TheHive, Draw.io, CyberChef, Mumble, Beats, Kibana, and Portainer.

Documentation

- CAPES <https://capesstack.io/>
- CAPES Documentation <https://github.com/capesstack/capes-docs>

▪ Cuckoo Sandbox

Description

- Cuckoo Sandbox is a free open-source sandbox to perform automated malware analysis.
- Cuckoo Sandbox is still being maintained, but updates are released slowly.

Documentation

- Cuckoo Sandbox Book <https://cuckoo.sh/docs/>

Training

- Josh Stroschen Current Malware Analysis Playlist <https://www.youtube.com/playlist?list=PLHJns8WZXCdueUdUTn-xw-eiBZuqSUGPG>
- Cuckoo Sandbox Overview and Demo <https://www.youtube.com/watch?v=V4z2tLRCuIY>

▪ Elastic

Description

- The Elastic Stack is a combination of Elasticsearch, Kibana, and Logstash (ELK) Stack. It is used to take data reliably and securely from any source, in any format, then search, analyze, and visualize it in real time.

Documentation

- Elastic Stack and Product Documentation <https://www.elastic.co/guide/index.html>

Training

- Elastic Blog <https://www.elastic.co/blog/>
- Elastic Webinars and Videos <https://www.elastic.co/videos/>

▪ Fast Incident Response

Description

- FIR (Fast Incident Response) is a cybersecurity incident management platform designed with agility and speed in mind. It allows for easy creation, tracking, and reporting of cybersecurity incidents.
- FIR is for anyone needing to track cybersecurity incidents (CSIRTs, CERTs, SOCs, etc.). It was developed as generic as possible before releasing it so that other teams around the world may also use it and customize it as they see fit.

Documentation

- FIR Github <https://github.com/certsocietegenerale/FIR>

▪ **Forensic Toolkit Imager**

Description

- Forensic Toolkit (FTK)® Imager is a data preview and imaging tool that lets operators quickly assess electronic evidence to determine if further analysis with a forensic tool, such as AccessData® FTK is warranted. FTK Imager can also create perfect copies (forensic images) of computer data without making changes to the original evidence.

Documentation

- FTK <https://accessdata.com/products-services/forensic-toolkit-ftk/ftkimager>
- Imager User Guide https://ad-pdf.s3.amazonaws.com/Imager/4_3_0/FTKImager_UG.pdf

Training

- DFIR Science Forensic Acquisition in Windows—FTK Imager https://www.youtube.com/watch?v=TkG4JqUcx_U

▪ **Google Rapid Response**

Description

- Google Rapid Response (GRR) is an incident response framework focused on remote live forensics. It consists of a python client (agent) that is installed on target systems, and python server infrastructure that can manage and talk to clients.
- The goal of GRR is to support forensics and investigations in a fast, scalable manner to allow analysts to quickly triage attacks and perform analysis remotely.

Documentation

- Github page <https://github.com/google/grr>
- Documentation website <https://grr-doc.readthedocs.io/en/latest/>

▪ **Grafana**

Description

- Grafana is an open-source platform that allows users to query, visualize, alert on, and understand metrics. Users are able to create, explore, and share dashboards.

Documentation

- Grafana github <https://github.com/grafana/grafana>
- Grafana Docs <https://grafana.com/docs/>

Training

- Grafana webinars and videos <https://grafana.com/videos/>

▪ **Kali**

Description

- Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing.

Documentation

- Kali Website <https://www.kali.org>
- Kali Documentation <https://www.kali.org/docs/>

Training

- Full Ethical Hacking Course - Network Penetration Testing for Beginners (2019) <https://www.youtube.com/watch?v=3Kq1MIfTWCE>

▪ MISP

Description

- A threat intelligence platform for gathering, sharing, storing, and correlating IOCs of targeted attacks, threat intelligence, financial fraud information, vulnerability information or even counter-terrorism information.

Documentation

- MISP Project Website <https://www.misp-project.org/>
- MISP Documentation <https://www.circl.lu/doc/misp/>

Training

- MISP General Usage Training Part 1 <https://www.youtube.com/watch?v=-NuODyh1YJE>
- MISP General Usage Training Part 2 <https://www.youtube.com/watch?v=LIKnh5b0bgw>

▪ Nmap

Description

- Nmap is a free and open-source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks but works fine against single hosts.

Documentation

- <https://nmap.org/>

Training

- Hackersploit YouTube tutorials <https://www.youtube.com/c/HackerSploit/featured>
- Nmap Tutorial for Beginners (Step by Step) | Nmap Vulnerability Scanning Guide https://www.youtube.com/watch?v=nJwZ2f_9rk

▪ OpenVAS

Description

- OpenVAS is a full-featured vulnerability scanner. Its capabilities include unauthenticated testing, authenticated testing, various high level, and low-level internet and industrial protocols, performance tuning for large-scale scans and a powerful internal programming language to implement any type of vulnerability test.

Documentation

- OpenVAS website <https://www.openvas.org/>
- GreenBone Github page <https://github.com/greenbone/openvas-scanner>

Training

- Hackersploit OpenVAS Setup and Configuration video <https://www.youtube.com/playlist?list=PLBf0hzazHTG0cYieYs4v-TV2amt2cYYaC>
- Hackersploit OpenVAS YouTube Playlist <https://www.youtube.com/playlist?list=PLBf0hzazHTGNxNNLU6eQ60JvTmlOD-QY9>

▪ **OSSEC Agent**

Description

- OSSEC (Open-Source Security) is a scalable, multi-platform, open-source Host-based Intrusion Detection System.

Documentation

- OSSEC Documentation <https://www.ossec.net/docs/>

▪ **Regshot**

Description

- Regshot is an open-source (GNU Lesser General Public License) registry compare utility that allows users to quickly take a snapshot of the registry and then compare it with a second one, done after doing system changes or installing a new software product.

Documentation

- How to Use Regshot to Monitor the Registry <https://www.howtogeek.com/198679/how-to-use-regshot-to-monitor-your-registry/>

▪ **REMnux**

Description

- REMnux® is a Linux toolkit for reverse-engineering and analyzing malicious software. REMnux provides a curated collection of free tools created by the community. Analysts can use it to investigate malware without having to find, install, and configure the tools.

Documentation

- REMnux Documentation <https://remnux.org/#docs>

Training

- Introduction to Malware Analysis <https://www.youtube.com/watch?v=f-fMdnUW4X4>

▪ **Security Onion**

Description

- Security Onion is a free and open Linux distribution for threat hunting, enterprise security monitoring, and log management. It includes TheHive, Playbook, Fleet, osquery, CyberChef, Elasticsearch, Logstash, Kibana, Suricata, Zeek, Wazuh, and many other security tools. Security Onion has been downloaded over 2 million times and is being used by security teams around the world to monitor and defend their enterprises.

Documentation

- Security Onion Documentation <https://docs.securityonion.net/en/2.3/index.html>

Training

- Security Onion Essential YouTube Playlist <https://www.youtube.com/watch?v=5fxVaVO8-L8&list=PLjFITO9rB155aYBjHw2lnKkSMLuhWpxH>

- **Signature-Base**

Description

- Signature-Base is a github repository that contains YARA signature and IOC database.

Documentation

- Github <https://github.com/Neo23x0/signature-base>

- **SIFT**

Description

- The SIFT Workstation is a collection of free and open-source incident response and forensic tools designed to perform detailed digital forensic examinations in a variety of settings. It can match any current incident response and forensic tool suite. SIFT demonstrates that advanced incident response capabilities and deep-dive digital forensic techniques can be accomplished using cutting-edge open-source tools that are freely available and frequently updated.

Documentation

- SIFT Workstation <https://www.sans.org/tools/sift-workstation/>

Training

- SIFT Workstation YouTube <https://www.youtube.com/c/SANSDigitalForensics/search?query=SIFT>

- **Sysinternals Suite**

Description

- Sysinternal are system utilities to help manage, troubleshoot, and diagnose Windows systems and applications.

Documentation

- Windows Sysinternals Documentation <https://docs.microsoft.com/en-us/sysinternals/>

- **Sysmon**

Description

- System Monitor (Sysmon) is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time. By collecting the events it generates using Windows Event Collection or SIEM agents and subsequently analyzing them, operators can identify malicious or anomalous activity and understand how intruders and malware operate on the network.

Note that Sysmon does not provide analysis of the events it generates, nor does it attempt to protect or hide itself from attackers.

Documentation

- Microsoft Sysmon <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
- Swift On Security Sysmon config <https://github.com/SwiftOnSecurity/sysmon-config>
- Olaf Hartong Sysmon Modular config <https://github.com/olafhartong/sysmon-modular>
- TrustedSec Sysmon Community Guide <https://www.trustedsec.com/tools/trustedsec-sysmon-community-guide/>

Training

- TrustedSec Sysmon YouTube Playlist https://www.youtube.com/playlist?list=PLk-dPXV5k8SFtMOngREKXCp8QyUnKI_I5

▪ **Suricata**

Description

- Suricata is independent open-source threat detection engine that is used as an IDS, IPS, and NSM tool.

Documentation

- <https://suricata.io/>

Training

- OISF-Suricata YouTube channel <https://www.youtube.com/c/OISFSuricata/featured>

▪ **TheHive**

Description

- A scalable, open-source and free Security Incident Response Platform, tightly integrated with MISP, designed to make life easier for SOCs, CSIRTs, CERTs and any information security practitioner dealing with security incidents that need to be investigated and acted upon swiftly.

Documentation

- TheHive <https://thehive-project.org/>
- TheHive Documentation <http://docs.thehive-project.org/thehive/>

Training

- TheHive Project YouTube Playlist <https://www.youtube.com/playlist?list=PLy-SBx6KOB-efrug9439Chopi5aFdWup>

▪ **ThreatIngester**

Description

- An extendable tool to extract and aggregate IOCs from threat feeds.

Documentation

- ThreatIngester Github <https://github.com/InQuest/ThreatIngester>
- ThreatIngester Documentation https://inquest.readthedocs.io/_/downloads/threatingstor/en/latest/pdf/

▪ **ThreatPinch Lookup**

Description

- A Chrome browser extension that enables users to plug API queries directly into the browser via an on-hover tooltip.

Documentation

- Github page with documentation <https://github.com/cloudtracer/ThreatPinchLookup>

Training

- ThreatPinch YouTube channel <https://www.youtube.com/channel/UCuhYal1qbb-exuhzscp3HBQ/featured>

- **Wireshark**

- Description*

- Wireshark is a widely used network protocol analyzer. It lets users see what's happening on the network at a microscopic level and is a standard tool used across many commercial and non-profit enterprises, government agencies, and educational institutions.

- Documentation*

- Wireshark User's Guide https://www.wireshark.org/docs/wsug_html_chunked/

- Training*

- Wireshark Training <https://www.wireshark.org/docs/>
 - NETRESEC Publicly available PCAP files <https://www.netresec.com/?page=PcapFiles>

- **Zeek**

- Description*

- Zeek is a passive, open-source network traffic analyzer. Many organizations use Zeek as a NSM to support investigations of suspicious or malicious activity. Zeek also supports a wide range of traffic analysis tasks beyond the security domain, including performance measurement and troubleshooting.

- Documentation*

- <https://docs.zeek.org/en/current/index.html>

- Training*

- Corelight YouTube channel has many videos on using Zeek <https://www.youtube.com/c/CorelightInc/featured>

Appendix D: Bibliography

- Bacon, M. (2015, October). *Indicators of Compromise (IOC)*. Retrieved March 4, 2021, from TechTarget
- Bianco, D. J. (2014). *The Pyramid of Pain*. Retrieved June 12, 2021, from https://rvasec.com/slides/2014/Bianco_Pyramid%20of%20Pain.pdf
- Caltagirone, S., Pendergast, A., & Betz, C. (2013). *The Diamond Model of Intrusion Analysis*. Retrieved April 30, 2021, from <https://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf>
- Cichonski, P., Millar, T., Grance, T., & Scarfone, K. (2012). *SP. 800-61 Rev 2: Computer Security Incident Handling Guide*. Retrieved July 28 2021, from <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>
- CIRCL. (2021). *MISP—Open Source Threat Intelligence Platform*. Retrieved July 28, 2021, from <https://www.circl.lu/services/misp-malware-information-sharing-platform/>
- CIRCL Luxembourg. (2018, November 6). *MISP Training Module 1—An Introduction to Cybersecurity Information Sharing*. Retrieved March. 3, 2021, from YouTube: <https://www.youtube.com/watch?v=aM7czPsQyal>
- CIRCL Team MISP Project. (n.d.). *MISP User Training—General Usage of MISP*. Retrieved April 1, 2021, from <https://www.misp-project.org/misp-training/1-misp-usage.pdf>
- Daszczyszak, R., Ellis, D., Luke, S., & Whitley, S. (2020, July). *TTP-Based Hunting*. Retrieved March 1, 2021, from <https://www.mitre.org/publications/technical-papers/ttp-based-hunting>
- Elastic. (n.d.). *Elastic Common Schema (ECS) Reference*. Retrieved July 22, 2021, from <https://www.elastic.co/guide/en/ecs/current/index.html>
- Elastic. (n.d.). *Lucene Query Syntax*. Retrieved July 22, 2021, from <https://www.elastic.co/guide/en/kibana/current/lucene-query.html>
- Faou, M., Tartare, M., & Dupuy, T. (2019, October). *Operation Ghost The Dukes Aren't Back—They Never Left*. Retrieved April 30, 2021, from https://www.welivesecurity.com/wp-content/uploads/2019/10/ESET_Operation_Ghost_Dukes.pdf
- FireEye. (2020, December 13). *Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor*. Retrieved April 23, 2021, from <https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html>
- FireEye. (n.d.). *Cyber Threat Intelligence 101*. Retrieved March 03, 2021, from <https://www.fireeye.com/mandiant/threat-intelligence/what-is-cyber-threat-intelligence.html>
- Google. (2020, November 30). *Stenographer*. Retrieved July 28, 2021, from <https://github.com/google/stenographer>
- Hartong, O. (2020). *Sysmon Modular*. Retrieved June 1, 2021, from <https://github.com/olafhartong/sysmon-modular/tree/version-12>
- Horovits, D. (2020, June 9). *The Complete Guide to the ELK Stack*. Retrieved July 28, 2021, from <https://logz.io/learn/complete-guide-elk-stack/#intro>
- Hutchins, E., Cloppert, M., & Amin, R. (n.d.). *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*. Retrieved July 30, 2021, from <https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf>
- Iklody, A., Alexandre, D., & CIRCL. (2018). Retrieved April 30, 2021, from <https://eugit.opencloud.lu/MISP/misp-objects>

JSON. (n.d.). *Introducing JSON*. Retrieved April 1, 2021, from <https://www.json.org/json-en.html>

Kent, K., & Souppaya, M. (2006, September). NIST SP 800-92: *Guide to Computer Security Log Management*. Retrieved April 30, 2021, from <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf>

Korban, C. A., Miller, P. D., Pennington, A., & Thomas, C. B. (2017). *APT3 Adversary Emulation Plan*. Annapolish Junction: The MITRE Corporation.

MISP. (2021, June 24). *MISP Objects*. Retrieved June 24, 2021, from <https://www.misp-project.org/objects.html>

MISP. (n.d.). *MISP*. Retrieved March 3, 2021, from <https://www.misp-project.org/index.html>

MISP. (n.d.). *MISP Galaxy Clusters*. Retrieved April 1, 2021, from www.misp-project.org/galaxy.html

MITRE. (2017, May 31). *APT29*. Retrieved April 30, 2021, from <https://attack.mitre.org/groups/G0016/>

MITRE. (2018, April 18). *Pupy*. Retrieved August 24, 2021, from <https://attack.mitre.org/software/S0192/>

MITRE. (2021). *Caldera*. Retrieved September 2, 2021, from <https://caldera.mitre.org>

MITRE. (2021). *Enterprise Techniques*. Retrieved 30 2021, April, from <https://attack.mitre.org/techniques/enterprise/>

MITRE. (n.d.). *Groups*. Retrieved July 28, 2021, from <https://attack.mitre.org/groups/>

NIST. (n.d.). Retrieved April 30, 2021, from Malicious Cyber Activity: https://csrc.nist.gov/glossary/term/malicious_cyber_activity

Nmap. (n.d.). *Introduction*. Retrieved July 28, 2021, from <https://nmap.org>

Nmap. (n.d.). *Legal Issues*. Retrieved July 28, 2021, from <https://nmap.org/book/legal-issues.html>

Palantir. (n.d.). *Alerting and Detection Strategies Framework*. Retrieved August 1, 2021, from <https://github.com/palantir/alerting-detection-strategy-framework>

Russinovich, M. (2021, August 18). *Autoruns for Windows v14.0*. Retrieved May 1, 2021

Russinovich, M., & Garnier, T. (2021, August 18). *Sysmon v13.24*. Retrieved August 18, 2021

Security Onion. (2021). *About Security Onion*. Retrieved July 28, 2021, from <https://docs.securityonion.net/en/2.3/about.html#security-onion>

Sophos. (2021, April 21). *How to Use Microsoft Autoruns to Locate Undetected Malware*. Retrieved April 22, 2021 , from https://support.sophos.com/support/s/article/KB-000035878?language=en_US

Sqrrl. (2015). *The Threat Hunting Reference Model Part 1: Measuring Hunting Maturity*. Retrieved July 30, 2021, from https://www.threathunting.net/files/The%20Threat%20Hunting%20Reference%20Model%20Part%201_%20Measuring%20Hunting%20Maturity%20_%20Sqrrl.pdf

Sqrrl. (n.d.). *Hunt Evil: Your Practical Guide to Threat Hunting*. Retrieved August 1, 2021, from <https://www.threathunting.net/files/hunt-evil-practical-guide-threat-hunting.pdf>

Suricata. (2019). *What is Suricata*. Retrieved July 28, 2021, from <https://suricata.readthedocs.io/en/suricata-6.0.3/what-is-suricata.html>

Zeek. (2021, August 20). *Zeek Documentation*. Retrieved July 28, 2021, from <https://docs.zeek.org/en/master/about.html>

Zeek. (n.d.). *Network Visibility*. Retrieved August 5, 2021, from <https://docs.securityonion.net/en/2.3/network.html>

Appendix E: Training Resources

The resources within this appendix are to consolidate the resources provided within this document for ease of use.

MITRE Resources

- MITRE TTP-Based Hunting: <https://www.mitre.org/publications/technical-papers/ttp-based-hunting>
- MITRE APT3 Emulation Plan: <https://attack.mitre.org/resources/adversary-emulation-plans/>
- MITRE APT29 Emulation Plan: https://github.com/mitre-attack/attack-arsenal/tree/master/adversary_emulation/APT29/Emulation_Plan
- MITRE ATT&CK: <https://attack.mitre.org>
- MITRE CAR: <https://car.mitre.org>
- MITRE ATT&CK Data Sources: <https://github.com/mitre-attack/attack-datasources>
- MITRE Caldera: <https://caldera.mitre.org>
- MITRE D3FEND™: <https://d3fend.mitre.org>
- MITRE Engage: <https://shield.mitre.org>
- MITRE CAPEC™: <http://capec.mitre.org/index.html>
- MITRE CWE™: <http://cwe.mitre.org>
- MITRE OVAL™: <http://oval.mitre.org>

Overview

- Prerequisite training
 - ELK or Security Onion: <https://www.youtube.com/watch?v=v69kyU5XMF4>
 - Suricata: https://www.youtube.com/results?search_query=training+on+suricata
 - Nmap: https://www.youtube.com/results?search_query=training+on+nmap
 - MITRE's Cyber Analytics Repository (CAR): https://www.youtube.com/results?search_query=mitre+cyber+analytics+repository
 - The MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK™) Framework: https://www.youtube.com/watch?v=kQIISQ4XR_Q
 - The MITRE ATT&CK Navigator: https://www.youtube.com/results?search_query=mitre+navigator
 - The MITRE ATT&CK Defender: <https://mitre-engenuity.org/mad/>

Prepare for the 7-step Process

- ADCS Tools
 - Security Onion Documentation: <https://docs.securityonion.net/en/2.3/about.html#security-onion>
 - Zeek Documentation: <https://docs.zeek.org/en/master/about.html>
 - Suricata Documentation: <https://suricata.readthedocs.io/en/suricata-6.0.3/what-is-suricata.html> <https://docs.securityonion.net/en/2.3/suricata.html>
 - Elasticsearch, Logstash, Kibana (ELK) Resource: <https://logz.io/learn/complete-guide-elk-stack/>
 - System Monitor (Sysmon) Documentation: <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
 - Malware Information Sharing Program (MISP) Documentation: <https://github.com/MISP/MISP>
 - Stenographer Documentation: <https://github.com/google/stenographer>
 - Network Mapper (Nmap): <https://nmap.org>

Step 1: Develop a Malicious Activity Model

- Creating a Malicious Activity Model
 - MITRE ATT&CK Groups: <https://attack.mitre.org/groups/>
- Leveraging Cyber Threat Intelligence
 - Alientvault: <https://otx.alienvault.com/>
 - CERT-EU: <https://cert.europa.eu/cert/filterededition/en/CERT-LatestNews.html>
 - CrowdStrike Blog: <https://www.crowdstrike.com/blog/>
 - Cybersecurity & Infrastructure Security Agency: <https://us-cert.cisa.gov>
 - ESET welvesecurity Blog: <https://www.welivesecurity.com/>
 - FireEye Blog: <https://www.fireeye.com/blog.html>
 - JPCERT: <https://blogs.jpCERT.or.jp/en>
 - Malpedia: <https://malpedia.caad.fkie.fraunhofer.de/actors>
 - Microsoft Security Intelligence: <https://www.microsoft.com/security/blog/microsoft-security-intelligence/>
 - OpenCTI: <https://www.opencti.io/en/>
 - Recorded Future: <https://www.recordedfuture.com/blog/>
 - Red Canary Blog: <https://redcanary.com/blog/>
 - Securelist: <https://securelist.com/>
 - SecureWorks Blog: <https://www.secureworks.com/blog>
 - Symantec Blog: <https://symantec-enterprise-blogs.security.com/blogs/>

- Talos Blog: <https://blog.talosintelligence.com/>
- ThaiCert Threat Actor Encyclopedia: <https://apt.thaicert.or.th/cgi-bin/aptgroups.cgi>
- The DFIR Report: <https://thedfirreport.com/>
- ThreatMiner: <https://www.threatminer.org/>
- Unit42: <https://unit42.paloaltonetworks.com/>
- Using Computer Incident Response Center Luxembourg's (CIRCL) Malware Information System Platform (MISP)
 - MISP Contact: <https://www.circl.lu/contact/>
 - MISP Download: <https://www.misp-project.org/download/>
 - MISP Training Modules: <https://www.misp-project.org/misp-training/1-misp-usage.pdf>
 - MISP README: <https://github.com/MISP/misp-objects/blob/main/README.md>
 - <https://www.misp-project.org/objects.html>
 - <https://www.misp-project.org/misp-training/1-misp-usage.pdf>
 - <https://www.misp-project.org/galaxy.html>

Step 2: Develop Hypotheses and Abstract Analytics

- Defining Hypothesis and Resources to Inform Hypothesis and Analytics
 - Adversary Emulation APT3: <https://attack.mitre.org/resources/adversary-emulation-plans/>
 - Adversary Emulation APT29: https://github.com/mitre-attack/attack-arsenal/blob/master/adversary_emulation/APT29/Emulation_Plan/APT29_EmuPlan.pdf
 - Azure Sentinel Hunting Queries: <https://github.com/Azure/Azure-Sentinel/tree/master/Hunting%20Queries>
 - Elastic Detection Rules: <https://github.com/elastic/detection-rules>
 - Event Query Language Analytics Library: <https://eqllib.readthedocs.io/en/latest/analytics.html>
 - Falcon Force Friday Github: <https://github.com/FalconForceTeam/FalconFriday>
 - MAGMA Use Case Framework: <https://www.betaalvereniging.nl/en/safety/magma/>
 - Palantir Alerting and Detections Strategies Framework: <https://github.com/palantir/alerting-detection-strategy-framework>
 - SOC Prime MITRE ATT&CK Map: <https://attack.socprime.com/#/>
 - Threat Hunter Playbook: <https://threathunterplaybook.com/introduction.html>
 - Uncoder.io: <https://uncoder.io/>
 - SigmaHQ: <https://github.com/SigmaHQ/sigma>

Step 3: Determine Data Requirements

- Leveraging ATT&CK Data Sources
 - MITRE ATT&CK: <https://attack.mitre.org>
 - MITRE CAR: <https://car.mitre.org>
 - MITRE ATT&CK Data Sources: <https://github.com/mitre-attack/attack-datasources>
- Determining Sysmon Data Requirements for T1053
 - Sysmon Community Guide: <https://github.com/trustedsec/SysmonCommunityGuide>
 - Sysmon Download: <https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon>
 - Olaf Hartong Sysmon Modular: <https://github.com/olafhartong/sysmon-modular>
- Taking Advantage of Additional Resources
 - Atomic Threat Coverage: <https://github.com/atc-project/atomic-threat-coverage>
 - Malware Archaeology Cheat Sheet: <https://static1.squarespace.com/static/552092d5e4b0661088167e5c/t/5b8f091c0ebbe8644d3a886c/1536100639356/Windows+ATT%26CK+Logging+Cheat+Sheet+ver+Sept+2018.pdf>
 - SwiftOnSecurity Sysmon Config: <https://github.com/SwiftOnSecurity/sysmon-config>

Step 4: Filtering Your Sources of Data

- MITRE Crown Jewels Analysis: <https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/crown-jewels-analysis>
- Understanding Time, Behavior, and Cyber Terrain
 - Living off the Land Techniques: <https://github.com/LOLBAS-Project/LOLBAS>
- Using Nmap and Zeek to Begin to Filter
 - Nmap Documentation: <https://nmap.org>
 - Nmap Legal Issues: <https://nmap.org/book/legal-issues.html>

Step 5: Identify and Mitigate Data Collection Gaps

- Security Onion Documentation: <https://docs.securityonion.net/en/2.3/pdf/>
- Identifying Data Using Security Onion
 - Suricata Rules: <https://suricata.readthedocs.io/en/suricata-6.0.3/rules/intro.html>
 - Zeek Logs: <https://docs.zeek.org/en/master/logs/index.html>
 - Zeek Log Fields: http://gauss.ececs.uc.edu/Courses/c6055/pdf/bro_log_vars.pdf
 - Zeek Custom Scripts: <https://docs.securityonion.net/en/2.3/zeek.html#custom-scripts>

Step 6: Test and Implement Analytics

- Implementing Pseudocode Analytics to Kibana
 - Elastic Prebuilt Rule Reference: <https://www.elastic.co/guide/en/security/current/prebuilt-rules.html>
- Exploring Adversary Emulation
 - <https://caldera.readthedocs.io/en/latest/Installing-CALDERA.html>
 - <https://caldera.readthedocs.io/en/latest/Plugin-library.html>
 - https://github.com/mitre-attack/attack-arsenal/tree/master/adversary_emulation/APT29
 - https://github.com/center-for-threat-informed-defense/adversary_emulation_library

Step 7: Hunt/Detect Malicious Activity and Investigate

- Responding to the Security Incident
 - Atomic Threat Coverage RE&CT Project: <https://atc-project.github.io/atc-react/>
 - Incident Response Consortium: <https://www.incidentresponse.com/playbooks/>
 - GuardSight, Inc. Playbooks Battle Cards: https://github.com/guardsight/gsvsoc_cirt-playbook-battle-cards
 - ElysiumSecurity Playbooks: <https://github.com/elysiumsecurityltd/IRM>
 - NIST SP 800-61: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>
- Additional Considerations
 - Palantir ADS Framework: <https://github.com/palantir/alerting-detection-strategy-framework>
 - Atomic Red Team: <https://github.com/redcanaryco/atomic-red-team>
 - MITRE Caldera: <https://caldera.mitre.org>

Appendix F: List of Figures

Figure 1. TTP-Based Hunt Methodology “V” Diagram	4
Figure 2. Pyramid of Pain (Bianco, 2014)	5
Figure 3. TTP-Based Hunt Methodology, with Mapped Tools	7
Figure 4. MITRE ATT&CK TTPs Network vs. Host Data Detection Capabilities	10
Figure 5. Simplified Network TAP Placement	12
Figure 6. Expanded Example of Network TAP placements	13
Figure 7. The Diamond Model	17
Figure 8. Log In for MISP	22
Figure 9. The State of the Art MISP Data Model	24
Figure 10. MISP Event Object with Attributes	25
Figure 11. Example of a MISP Object Template	26
Figure 12. Example Galaxies in MISP	27
Figure 13. Sample List of Events in MISP	28
Figure 14. Adding an Event in MISP	28
Figure 15. Adding Attribute Data to an Event in MISP	29
Figure 16. Categories in MISP	29
Figure 17. Example of How to Add Tags to an Event in MISP	30
Figure 18. Searching for APT29 in MISP	31
Figure 19. Searching for Scheduled Task Technique T1053 and APT29 in MISP	32
Figure 20. Searching for T1053 and APT29 in MISP	32
Figure 21. Correlated Data associated with Operation Ghost	33
Figure 22. Cover of Operation Ghost White Paper	33
Figure 23. Screen Capture of MITRE ATT&CK Techniques	35
Figure 24. Screen Capture from Operation Ghost White Paper	35
Figure 25. Operation Ghost White Paper Correlated Events	36
Figure 26. Correlated Data to the MITRE ATT&CK for Enterprise Framework	37
Figure 27. MITRE ATT&CK Matrix	38
Figure 28. CAR Analytics List	41
Figure 29. CAR Analytics List, Organized by ATT&CK Technique	41
Figure 30. ATT&CK—Scheduled Task Analytics	42
Figure 31. CAR Analytic—Scheduled Task	43
Figure 32. APT29 Adversary Emulation Plan—ATT&CK Technique	44
Figure 33. ATT&CK Data Sources	49
Figure 34. ATT&CK Website—T1053.005 Data Sources	49

Figure 35. Example Data Component for T1053.005	50
Figure 36. Sysmon Install Package	51
Figure 37. CAR-2013-08-00 Analytic Implementation Pseudocode	51
Figure 38. Sysmon Configuration Excerpt	52
Figure 39. Elements of Time, Behavior, and Cyber Terrain	55
Figure 40. Security Onion Dashboard	61
Figure 41. Security Onion Alerts	61
Figure 42. Alert Options	62
Figure 43. Alert PCAP	62
Figure 44. Security Onion Hunt	63
Figure 45. Search Options	63
Figure 46. PCAP	64
Figure 47. PCAP Stream	65
Figure 48. PCAP Detailed	65
Figure 49. Security Onion CyberChef	66
Figure 50. Security Onion Grafana	66
Figure 51. Security Onion TheHive	67
Figure 52. Suricata Rule	67
Figure 53. Strelka	69
Figure 54. Fleet Interface	69
Figure 55. Fleet View	70
Figure 56. Fleet Query	70
Figure 57. Security Onion 2 Network Flow Diagram	71
Figure 58. So Zeek Logs	72
Figure 59. Examples of Zeek Logs	72
Figure 60. Example Zeek conn.log	73
Figure 61. Kibana Main Page	74
Figure 62. Index Management	75
Figure 63. CAR-2013-08-001: Execution with Scheduled Tasks	79
Figure 64. Kibana Query for Scheduled Tasks	79
Figure 65. NIST Incident Response Lifecycle	83
Figure 66. General Hunt Process Flow	85
Figure C-1. APT28 ATT&CK Mapping	108
Figure C-2. APT29 ATT&CK Mapping	109

Appendix G: List of Tables

Table 1. Pyramid of Pain IOCs	6
Table 2. Sample Malicious Activity Model for APT29	18
Table 3. Cyber Hunt Plan	19
Table 4. Cyber Hunt Plan	31
Table 5. Cyber Hunt Plan Update—Hypotheses & Abstract Analytics	45
Table 6. Cyber Hunt Plan Update—Determine Data Requirements	52
Table 7. Information to Acquire from Network Owner	54
Table 8. Cyber Hunt Plan	57
Table 9. Using Nmap to Filter	58
Table 10. Cyber Hunt Plan	78
Table 11. Cyber Hunt Plan	82
Table 12. Threat Hunting Techniques	86
Table 13. Cyber Hunt Report Example	96
Table 14. Use Case Framework	100
Table A-1. APT ATT&CK Techniques	105
Table B-1. APT28 MITRE ATT&CK and Data Sensor Mapping	108
Table B-1. APT29 MITRE ATT&CK and Data Sensor Mapping	109
Table C-1. Discovery Tools	110
Table C-2. Network-Based DFIR	110
Table C-3. Host-Based DFIR	110
Table C-4. Vulnerability Assessment Tools	111
Table C-5. Mitigation, Clearing, and Remediation Tools	111
Table C-6. Forensics and Malware Analysis Tools	111
Table C-7. Continuous Monitoring Tools	112
Table C-8. Event Correlation & Analysis Tools	112
Table D-9. Threat Emulation Tools	112
Table C-10. Cyber Threat Intelligence (CTI) Tools	113
Table C-11. Suite of Tools	113
Table C-12. Administrative Tools	113

Appendix H: Abbreviations and Acronyms

ADCS	Active Defense Capability Set
ADS	Alerting Detection Strategies
APT	Advanced Persistent Threat
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
C2	Command and Control
CAPEC	Common Attack Pattern Enumeration and Classification
CAR	Cyber Analytics Repository
CERT	Computer Emergency Response Team
CIDR	Classless Inter-Domain Routing
CIRCL	Computer Incident Response Center Luxembourg
COI	Community of Interest
CPU	Central Processing Unit
CSIRT	Computer Security Incident Response Teams
CTI	Cyber Threat Intelligence
CWE	Common Weakness Enumeration
D3FEND	Detection, Denial, and Disruption Framework Empowering Network Defense
DCO	Defensive Cyber Operations
DFIR	Digital Forensic and Incident Response
DNS	Domain Name Server
ECS	Elastic Common Schema
ELK	Elasticsearch, Logstash, and Kibana
FIR	Fast Incident Response
FTK	Forensic Toolkit
FTP	File Transfer Protocol
Gb	Gigabyte
GRR	Google Rapid Response
GUI	Graphical User Interface
HMM	Hunt Maturity Model
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
ID	Identification
IDS	Intrusion Detection System
IOC	Indicators of Compromise
IP	Internet Protocol
IPS	Intrusion Prevention System
JSON	JavaScript Object Notation
KQL	Kibana Query Language

LSASS	Local Security Authority Server Service
MD5	Message Digest 5
MISP	Malware Information Sharing Program
N/A	Not Applicable
NATO	Northern Atlantic Treaty Organization
NIC	Network Interface Controller
NIST	National Institute of Standards and Technology
Nmap	Network Mapper
NSM	Network Security Monitor
OISF	Open Information Security Foundation
OSINT	Open-Source Intelligence
OSSEC	Open-Source Security
OSSEM	Open-Source Security Events Metadata
OTRF	Open Threat Research Forge
OVAL	Open Vulnerability and Assessment Language
PCAP	Packet Capture
RDP	Remote Desktop Protocol
Regex	Regular Expression
Schtask	Scheduled Tasks
SHA	Secure Hash Algorithm
SIEM	Security Incident and Event Management
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SOC	Security Operations Center
SPAN	Switchport Analyzer
SQL	Structure Query Language
SYN	Synchronization
Sysmon	System Monitor
TAP	Test Access Point
TCP	Transfer Control Protocol
TTP	Tactics, Techniques, and Procedures
UDP	User Datagram Protocol
USEUCOM	United States European Command
WMI	Windows Management Instrumentation
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

ABOUT THE AUTHORS

Travis Gloor is a Lead, Cyber Assessment Engineer employed by MITRE on June 14th, 2021, in Division L521, Cyber Assessment. Travis has 14 year of Information Technology experience with 7 years being cybersecurity focused for different DoD organizations, along with a Master Information Assurance. His experience ranges from being an Enterprise Administration to Cyber Threat Hunting. He worked with the Missile Defense Agency's Cyber Assistance Team to improve detection techniques and processes.

Eric Hazard is a Senior Cybersecurity Engineer with the MITRE Corporation. Eric originally joined MITRE in 2014 as an intern and at the completion of his undergraduate degree in Computer Security and Information Assurance from Norwich University (Vermont), Eric joined MITRE full time in 2017. Eric initially joined MITRE's Cyber New Professional program where he supported Digital Forensics, Defensive Cyber Operations, and Mission Analysis projects. Since matriculating into MITRE's L522 Cyber Resiliency department, part of MITRE Labs, Eric has supported partner nation capacity building, the development of a Digital Forensic standard, and US European Command. Eric currently is supporting multiple partner nation capacity building projects while pursuing a master's degree in Cybersecurity Policy and Governance at Boston College.

Ronald Mercado is a Senior Cybersecurity Engineer employed with MITRE for 6 months joining in May 2021. He has 9 years of Information Technology experience with 6 years being Cybersecurity focused, holds a Masters in Cybersecurity and Information Assurance, and several industry certifications. Ronald is in the department L511 Defensive Cyber Operations focusing on developing effective cyber detection, analysis, and engineering solutions to support defensive operational capabilities on deep understanding of the threat. Ronald has been able to focus on threat hunting operations and maturing those processes for sponsor programs. Previous roles have been a SIEM engineering, Splunk administration, and Threat Hunter for industry and Department of Defense.

Denise Olsen is a Principal Engineer, Cyber Operations with the MITRE Corporation. She has over 20 plus years of experience in both the Information Technology and Cyber domain and holds a Master of Science, Information Systems from Colorado Technical University and graduated from the University of Texas at Austin, Senior Executive Services fellowship program. Denise is currently in MITRE's Joint Staff and Combatant Command Division, Europe and Africa Regional Operations providing support to United States European Command (USEUCOM), Joint Cyber Center as a subject matter expert (SME) in Cyber Operations. She also provides SME support to US Africa Command and MITRE's National Security Sector International Division in the development of the Active Defense Capability Set project that will enable developing partner nation Cyber Forces in the USEUCOM and US Africa Combatant Command area of responsibility to master the skills of Tactics, Techniques, and Procedures Based Cyber Hunting.

ABOUT MITRE

MITRE's mission-driven teams are dedicated to solving problems for a safer world. Through our public-private partnerships and FFRDCs, we work across government and in partnership with industry to tackle challenges to the safety, stability, and well-being of our nation.

MITRE

SOLVING PROBLEMS
FOR A SAFER WORLD®