Lateral movement detection using ELK stack

A Thesis

Presented to

the Faculty of the Department of Information and Logistics Technology

University of Houston

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science Information System Security

By

Utkarsh Jain

May 2018

Lateral movement detection using ELK stack

Utkarsh Jain

APPROVED:

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

Chris Bronk, PhD
Assistant Professor
Information and Logistics Technology

Denise M Kinsey, PhD
Assistant Professor
Information and Logistics Technology

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

Dan Cassler
Interim Chair, Department of
Information and Logistics Technology

Abstract

In recent time, it is becoming increasingly difficult to prevent initial infiltration of a network, making it important to always consider and improve detection of an ongoing incident. Any attacker spends most energy, time and is additionally the most defenseless against detection when the attacker is trying to move laterally from system to system to escalate privileges. This research takes into consideration, majority of tools and techniques used by adversaries to move in a windows-based network laterally. It showcases the execution of these tools and suggests how usage of such tools can be detected using logging. Hence, this research is an attempt at creating a solution which helps in detection of lateral movement happening inside a windows-based network by collecting logs and analyzing them using ELK stack as the logging tool.

Table of Contents

Contents

Table of Figures

Introduction

Just a few short years ago, a sophisticated data breach was newsworthy, not in light of the harm done, but rather in view of how uncommon it was for a technically capable hack to happen. Not anymore, advanced cyber threats have come of age. The 'kill chain' of the advanced hacker has developed. (Alert Logic Staff, 2016) A brisk passage through spear phishing, web application defects, or misconfigurations, trailed by a precise and systematic compromise of multiple assets, lastly, accomplishing the mission objective.

Despite what most may expect, the most critical phase of an attack isn't the initial exploit, or the final information exfiltration. It's the middle phase, where the attacker seeks out assets, gains additional privileges and silently moves from system to system, subnet to subnet, nearer to his last objective. This stage is called *lateral movement*, and it is where the attacker invests the most energy and time and is additionally the most defenseless against detection.

Researches demonstrate that 80% of time in an attack is spent amid lateral movement. (SMOKESCREEN, 2016) The real breach happens reasonably quickly, and the last objective is quickly accomplished as well. It's moving from initial breach to the last objective that takes hackers time and resources. So, an approach needs to be developed to quickly detect these critical events to respond. If the attacker gets caught moving laterally in the network, it's game over for the attacker.

Hence, this research aims at creating a method to detect lateral movement happening inside a network by collecting logs and analyzing them for events which show lateral movement occurring inside the network. There are various techniques and tools associated with lateral movement, making it important to study these in detail to come up with a solution to detect them.

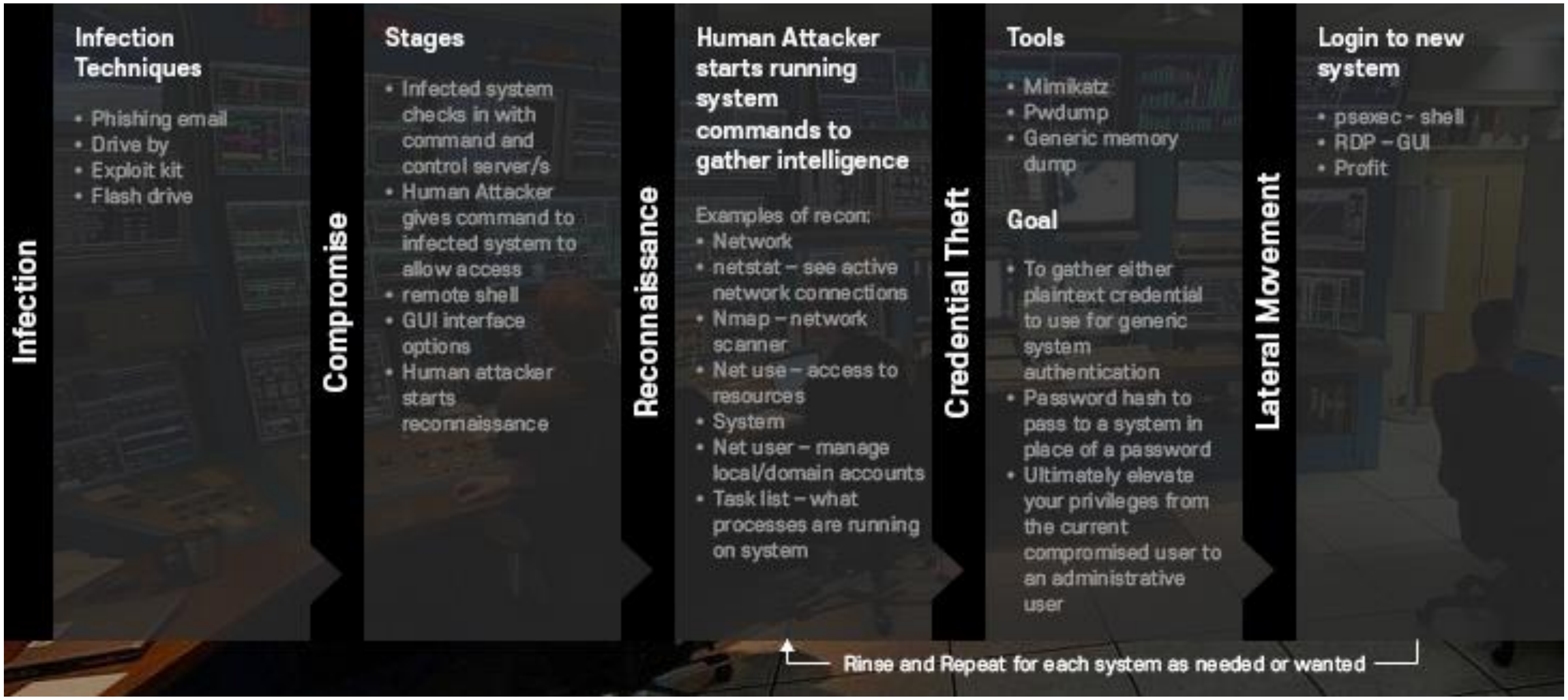
This research will explain various techniques and some commonly used tools by adversaries to perform lateral movement. ELK stack will be used as the tool for collecting and analyzing logs of the network because it is a free, easily available and easy to configure tool. The scope of this research only covers windows environment-based networks and selective ways of lateral movements. Various test cases will be generated to document the usage of these commonly used tools. Logs will be captured and analyzed for each of these test cases which will help in developing an understanding of what to look for. These documented cases will then help in creation of filters in ELK stack, hence, forming a way to detect the usage of these specific tools for performing lateral movement.

This research document, firstly describes what is lateral movement and various techniques of performing lateral movement. Then ELK stack will be discussed in detail along with its technical specifications, basic implementation and configuration. This will be followed by the chapter of PowerShell logging which will explain the setup and configuration required to collect PowerShell logs on ELK stack. After developing a detailed understanding about lateral movement and ELK stack, test cases involving the usage of various tools used to perform lateral movement will be discussed. These test cases will help in creating a method to detect lateral movement using ELK stack which will be discussed in the last section.

Lateral movement

Lateral movement can be defined as "Movement across a network from one system to another utilizing techniques and tools that enable an attacker to access and/or control systems within the organization's environment, which is necessary to achieve an adversary's goals." (MITRE, 2017) (Nolette, 2017)Lateral Movement is a critical step in the process of carrying out an attack on a network. Lateral movement techniques are generally utilized as a part of modern sophisticated cyber-attacks, specifically in Advanced Persistent Threats (APTs). (M.SORIA-MACHADO, 2017)

The adversary utilizes these techniques to access other hosts from a compromised system and get access to sensitive resources, such as mailboxes, shared folders, or credentials. These can be used in turn for compromise of additional systems, privilege escalation, or stealing more valuable credentials. This sort of attack may eventually offer access to the Domain Controller and give full control of a Windows-based framework or business-related administrator accounts. The capacity to remotely execute scripts or code can be a foundation of an attack, however adversaries also attempt to reduce their footprint in environments by abusing legitimate credentials combined with native network and OS functionality to remotely access systems. Figure 1 explains the processes involved in lateral movement.



Infection Techniques

- Phishing email
- Drive by
- Exploit kit
- Flash drive

Stages

- Infected system checks in with command and control server/s
- Human Attacker gives command to infected system to allow access
- remote shell
- GUI interface options
- Human attacker starts reconnaissance

Human Attacker starts running system commands to gather intelligence

Examples of recon:
- Network
- netstat – see active network connections
- Nmap – network scanner
- Net use – access to resources
- System
- Net user – manage local/domain accounts
- Task list – what processes are running on system

Tools

- Mimikatz
- Pwdump
- Generic memory dump

Goal

- To gather either plaintext credential to use for generic system authentication
- Password hash to pass to a system in place of a password
- Ultimately elevate your privileges from the current compromised user to an administrative user

Login to new system

- psexec - shell
- RDP – GUI
- Profit

Rinse and Repeat for each system as needed or wanted

Figure 1 : Lateral movement process chart (Nolette, 2017)

Different Techniques used for Lateral Movement

Understanding different techniques and tools used by adversaries for performing lateral movement is vital to build test cases which will form the basis of detection of these lateral movments. An adversary can utilize lateral movement for many reasons, including remote Execution of tools, pivoting to additional systems, access to specific information or files, access to additional credentials, or to cause an impact. Lateral movement, and the techniques that lateral movement depends on, are often vital to an adversary's set of capabilities and part of a more extensive set of information and access dependencies that the adversary takes advantage of within a network. To comprehend inherent security conditions, it is essential to know the relationships between accounts and access privileges across all systems on a network.

Following, are various Lateral Movement techniques used according to requirement (SMOKESCREEN, 2016) (MITRE, 2017):

Using Psexec

Psexec is a system system administration tool, created by Sysinternals before their acquisition by Microsoft. Psexec and the whole pstools suite lets administrators to remotely control Windows systems from the terminal. (SMOKESCREEN, 2016) Attackers cherish psexec for its capacity of uploading, executing and collaborating with an executable on a remote host. Since its additionally a legitimate system administration tool, it is invariably not blacklisted or identified by antivirus solutions. Psexec works from a command line, is effortlessly scriptable, and doesn't caution the remote client to its operation making it difficult to detect.

Exploitation of Remote Services

A shared objective for post-compromise exploitation of remote services is for lateral movement to enable access to a remote system. An adversary may need to decide whether the remote system is in a vulnerable state, which might be done through network service scanning or other discovery techniques searching for common, vulnerable software that may be deployed in the network, the absence of certain patches that may demonstrate vulnerabilities, or security software that might be utilized to identify or contain remote exploitation. Servers are likely a high value target for lateral movement exploitation, but endpoint systems may also be at risk if they provide an advantage or access to additional resources. There are a few well-known vulnerabilities that exist in common services such as SMB and RDP as well as applications that might be utilized inside internal networks such as MySQL and web server services. (MITRE, 2017)

Dependent upon the authorization level of the vulnerable remote service an adversary may accomplish exploitation for privilege escalation because of lateral movement exploitation too. An adversary may utilize valid accounts to sign into a service particularly intended to acknowledge remote connections, for example, telnet, SSH, and VNC. The adversary may then perform activities as the signed-on client.

Remote desktop protocol

Remote desktop is a feature in operating systems which enables a client to log into an interactive session with a system desktop graphical user interface on a remote system. Adversaries may associate with a remote system over RDP/RDS to expand access if the service is enabled and enables access to accounts with known credentials. Adversaries will probably utilize credential access techniques to acquire credentials to use with RDP.

When attackers have substantial certifications, RDP is the weapon of choice to increase interactive access to the assets. Since RDP sessions are encrypted, they're obscure to observing solutions (which regardless would not flag them as they are such a common legitimate administrative mechanism). Adversaries may perform RDP session capturing which includes taking a genuine client's remote session. Typically, a client is told when another person is trying to take their session and prompted with a question. With System permissions and utilizing Terminal Services Console, an adversary can seize a session without the requirement for credentials or prompts to the user. It can also lead to remote system discovery and privilege escalation by stealing a higher privileged account session.

Remote registry

Windows Registry is the core of the operating system. While generally utilized as a major aspect of a larger technique, the capacity to remotely control the Windows registry can be utilized to incapacitate security systems, remove auto-start programs and services, and install persistence mechanisms.

Windows Admin Shares

Windows systems have hidden network shares that are open just to administrators and give the capacity to remote access every disk volume on a network, to perform remote file copy and other administrative functions. These shares may not be permanently deleted but may be disabled. Adversaries may utilize this strategy in conjunction with admin level valid accounts to remotely get to a networked system over server message block (SMB) to interface with systems utilizing remote procedure calls (RPCs), transfer files, and run transferred binaries through remote

Execution. Example execution techniques that rely on authenticated sessions over SMB/RPC are Scheduled Task, Service Execution, and Windows Management Instrumentation. Adversaries can likewise utilize NTLM hashes to access administrator shares on systems with Pass the Hash and certain configuration and patch levels. (MITRE, 2017)

Windows Remote Management

Windows Remote Management (WinRM) is a component of the Windows Hardware Management features that manage server hardware locally and remotely. Windows Remote Management (WinRM) is the name of both a Windows service and a protocol that enables a client to collaborate with a remote system. The service implements the WS-Management protocol, hardware diagnosis and control through baseboard management controllers (BMCs). WinRM can also be used for scripting objects that allow to write applications that communicate remotely through the WS-Management protocol.

PowerShell

PowerShell command-line shell designed for task automation and configuration management framework which built on top of the .NET Framework common language runtime (CLR). It includes an interactive prompt and a scripting environment that can be used independently or in combination. PowerShell is an extremely powerful object-oriented scripting facility is built-in to every modern version of Windows, which can be used to steal in-memory credentials, modify system configuration and automate movement from one system to the next. PowerShell also provides a hosting API with which the PowerShell runtime can be embedded inside other applications. These applications can then use PowerShell functionality to implement certain operations. It also provides full access to COM and WMI, enabling administrators to perform administrative tasks on both local and remote Windows systems.

Port-scans

The basic port-scan, perhaps the main method that has remained virtually unaltered from the days when hacking started, is utilized to quickly identify services of interest — typically web applications, database servers and remote access functionality. While a full-blown port-scan is easily detected, “low and slow” scans get past practically any network monitoring system. Just simple TCP connects are sufficient for finding targets.

WMI

The Windows Management Instrumentation framework is Microsoft’s built-in system to manage the configuration of Windows systems. WMI scripts can be written automate administrative tasks on remote computers. WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components.

WMI can be utilized to start remote processes, query system information, or even, as has recently been demonstrated, store persistent malware that does not touch the disk in a conventional sense. Attackers make broad utilization of WMI as a method for rapidly listing system data to classify targets.

Scheduled tasks

Task Scheduler is a component of Microsoft Windows that provides the ability to perform routine tasks on a chosen computer. Just by monitoring a chosen criterion, it can schedule the launch of programs or scripts at pre-defined times or after specified time intervals. In terms of performing lateral movement, this can be called any attackers favorite technique.

The basic Windows 'at' command allows an attacker to schedule a task to execute on either a local or remote system. This capacity isn't utilized only to time execution, in many circumstances, scheduled tasks will run as the SYSTEM user, giving an attacker a chance to escalate privileges to complete control of the host on which the scheduled task runs. It's additionally an incredible method to schedule batch jobs that may utilize CPU or bandwidth, for example, compressing folders and transferring them over the system. By scheduling the task out of office hours, there is lesser possibility of discovery.

Token stealing

Windows uses access tokens to determine the ownership of a running process. Usage of built-in Windows API functions to copy access tokens from existing processes is known as token stealing. (MITRE, 2018) Manipulating access tokens is the easiest way to make a running process appear as though it belongs to someone other than the user that started the process. Being a recent technique in the public domain, stealing tokens from memory has turned into extremely popular technique, and is utilized in almost every attack nowadays. Tools, for example, mimikatz and Windows Credential Editor can discover service accounts in memory, create Kerberos tickets and elevate an attacker from ordinary client to domain administrator in almost no time. Adversaries may use access tokens to operate under a different user or system security context to perform actions and evade detection. An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing.

Pass the hash

Because of how the NTLM protocol functions, attackers can utilize the encrypted hash of a password to authenticate to remote services without knowing what the plaintext password is. This technique sidesteps standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash. After obtaining the password hashes, the attacker simply passes them on to other services, without having to undertake dictionary or brute-force attacks on the hash itself. Once validated, PtH (Pass the hash) might be utilized to perform activities on local or remote systems. This technique has been superseded in many attacks by token stealing but has still been used to devastating effect in recent breaches, for example, the Target Corporation attack. (SMOKESCREEN, 2016)

Pass the Ticket

Kerberos authentication can be utilized as the initial step to lateral movement to a remote system. (MITRE, 2017) Pass the ticket (PtT) is a method of authenticating to a system utilizing Kerberos tickets without having access to an account's password. In this technique, valid Kerberos tickets for valid accounts are captured by credential dumping. A user's service tickets or ticket granting ticket (TGT) may be obtained, depending on the level of access. A service ticket allows for access to a resource, whereas a TGT can be used to request service tickets from the Ticket Granting Service (TGS) to access any resource the user has privileges to access. Silver Tickets can be acquired for services that use Kerberos as an authentication mechanism and are used to generate tickets to access that resource and the system that hosts the resource (e.g., SharePoint). Golden Tickets can be acquired for the domain utilizing the Key Distribution Service account KRBTGT account NTLM hash, which enables generation of TGTs for any account in Active Directory. (MITRE, 2017)

Active directory

Active Directory is a directory service developed by Microsoft for Windows domain networks, and practically anything of value has a place in it. It's expected that the main thing attackers go for is the list of AD computers. The names in this list are the starting point for categorizing targets. A quick search for databases, backup systems, SCADA controllers, they read the names. Moreover, breaking into the domain controller or gaining domain administrator privileges gives hackers everything they need, at least as far as Windows environments are concerned.

Stolen credentials

Gaining legitimate credentials is easier once on the internal network, and once the attacker has them, it's safer for the attacker to utilize legitimate administration channels and stolen credentials to complete their mission. Legitimate credentials can be stolen, phished, cracked, key logged or gained in some other fashion. Notably, the FIN-4 financial APT group ran a highly successful campaign against Wall Street using nothing but stolen credentials. (SMOKESCREEN, 2016)

Breached host analysis

At the point when attackers breach an initial host, they plunder it for data that can enable them to move further. This incorporates passwords in text files, details of other systems, operating procedures, and even screen captures of how the user is working. They'll even just figure out the internal hierarchy and politics of the organization to craft social engineering attacks. This can be considered to be foot printing behind the firewall, when properly done, it sets the stage for devastating attacks that can lead to a cascade of compromised systems.

Network sniffing

Network sniffing involves a hardware device or a separate software program or a combination of both that enable real-time monitoring and analysis of data packets flowing over computer networks. Basically, it examines traffic on the network and takes snapshot copies of the packet data. While switched networks have made promiscuous mode sniffing less of an issue, attackers still gain tremendous value from setting up network sniffing on a high-traffic server to gain access to credentials of customers and other information. User segments are usually subjected to man-in the middle attacks like ARP spoofing, explained below.

ARP spoofing

ARP spoofing, is a technique by which an attacker sends falsified ARP (Address Resolution Protocol) messages onto a local area network. This results in the linking of an attacker's MAC address with the IP address of another host, such as the default gateway, causing any traffic meant for that IP address to be sent to the attacker instead. ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. It is one of the oldest techniques used, however, it's still utilized. Producing gratuitous fake ARP requests and replies can let attackers interject themselves in communications in switched networks in a classic man-in-the-middle attack. While these attacks have dropped out of support to some degree, they're not identified on numerous networks, and can be to an extremely damaging if an attacker finds the correct hosts to harm.

Email pillaging

Gaining access to email inboxes, either on the workstations, the server, or through webmail gives an attacker tremendous leverage. Besides the obvious information gathering and second tier spear phishing attacks, there are instances of attackers watching an incident response process being discussed over email and modifying their tactics accordingly. (SMOKESCREEN, 2016)

Application Deployment Software

Access to a network-wide or enterprise-wide software deployment system empowers an adversary to have remote code execution on all systems that are associated with such a system. The access might be utilized to laterally move to systems, accumulate data, or cause a particular impact, for example, wiping the hard drives on all endpoints. Adversaries may send malicious software to systems within a network utilizing application deployment systems employed by enterprise administrators. The authorizations required for this action vary by system configuration; local credentials might be adequate with direct access to the deployment server, or specific domain credentials might be required. Nonetheless, the system may require an administrative account to log in or to perform software deployment.

Distributed Component Object Model

Windows Distributed Component Object Model (DCOM) is a set of Microsoft concepts and program interfaces used for communication between software components on networked computers. It is straightforward middleware that broadens the usefulness of Component Object Model (COM) past a local computer using remote procedure call (RPC) technology. COM is a part of the Windows application programming interface (API) that empowers connection between software objects. Through COM, a client object can call methods of server objects, which are typically Dynamic Link Libraries (DLL) or executables (EXE). Permissions to interact with local and remote server COM objects are specified by access control lists (ACL) in the Registry. By default, only Administrators may remotely activate and launch COM objects through DCOM.

Adversaries may utilize DCOM for lateral movement. Through DCOM, adversaries operating in the context of an appropriately privileged user can remotely acquire arbitrary and even direct shellcode execution through Office applications as well as other Windows objects that contain insecure methods. DCOM can also execute macros in existing documents and may also invoke Dynamic Data Exchange (DDE) execution directly through a COM created instance of a Microsoft Office application, bypassing the need for a malicious document. (MITRE, 2017)

Logon Scripts

A login script is a series of instructions that a workstation follows every time a user logs on. These instructions are held on the server in a batch file that workstations can access and run. Windows permits logon scripts to be run whenever a specific user or group of users log into a system. The scripts can be utilized to perform administrative functions, which may often execute other programs or send data to an internal logging server.

On the off chance that adversaries can get to these scripts, they may embed extra code into the logon script to execute their tools when a client signs in. This code can enable them to maintain persistence on a single system, if it is a local script, or to move laterally within a network if the script is stored on a central server and pushed to many systems. Depending on the access configuration of the logon scripts, either local credentials or an administrator account may be necessary. (MITRE, 2018)

Replication Through Removable Media

Adversaries may use autorun features when the media is inserted into a system and executes, to move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media. In the case of Lateral Movement, this may occur through modification of executable files stored on removable media or by copying malware and renaming it to look like a legitimate file to trick users into executing it on a separate system. In the case of Initial Access, this may occur through manual manipulation of the media, modification of systems used to initially format the media, or modification to the media's firmware itself. (MITRE, 2018)

SSH Hijacking

Secure Shell (SSH) is a network protocol that provides administrators with a secure way to access a remote computer. SSH also refers to the suite of utilities that implement the protocol. Secure Shell provides strong authentication and secure encrypted data communications between two computers connecting over an insecure network such as the Internet. It enables a user to connect with another system by means of an encrypted tunnel, usually authenticating through a password, certificate or the utilization of an asymmetric encryption key pair. SSH is widely used by network administrators for managing systems and applications remotely, allowing them to log in to another computer over a network, execute commands and move files from one computer to another.

To move laterally from a compromised host, adversaries may take advantage of trust relationships established with other systems via public key authentication in active SSH sessions by hijacking an existing connection to another system. This may occur through compromising the SSH agent itself or by having access to the agent's socket. If an adversary can obtain root access, then hijacking SSH sessions is likely trivial. Compromising the SSH agent also provides access to intercept SSH credentials. (MITRE, 2017)

Taint Shared Content

As a collaboration mechanism, file shares are used both on central file servers, as well as by individual users. They often contain customer databases, details of additional systems, operating procedures, and useful software. Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files. Once a user opens the shared tainted content, the malicious portion can be executed to run the adversary's code on a remote system. Adversaries may use tainted shared content to move laterally or to elevate privileges. (MITRE, 2017)

Third-party Software

The VNC GUI remote access is still used in thousands of companies for remote technical support. VNC or other third-party applications and software deployment systems may be in use in the network environment for administration purposes. If an adversary gains access to these systems, then they may be able to execute code. All these types of software are extensively used by attackers who target one administrator with legitimate access, and then use those credentials to access all the other systems in the environment. Adversaries may gain access to and use third-party application deployment systems installed within an enterprise network. Access to a network-wide or enterprise-wide software deployment system enables an adversary to have remote code execution on all systems that are connected to such a system. The access may be used to laterally move to systems, gather information, or cause a specific effect, such as wiping the hard drives on all endpoints.

ELK stack

ELK stack is an end to end solution for deep search, data analytics, centralized logging and visualization. In this research, ELK stack will be used as the tool for collecting and analyzing logs of the network because it is a free, easily available and easy to configure tool. The components of ELK stack are, Elasticsearch, Logstash and Kibana:

Elasticsearch:

Elasticsearch (which is the E in the ELK Stack) is an open-source search engine based on Lucene. Elasticsearch is developed in Java and is accessible using an extensive and elaborate API (Application Programming Interface). It can also power very fast searches that are able to support customer data discovery applications. Elasticsearch represents the core of the ELK Stack (or BELK which stands for Elasticsearch, Logstash, Kibana and Beats). (Zarzosa, 2017)

Logstash:

Logstash (which is the L in the ELK Stack) is an open-source server-side data processing pipeline tool. One of its main scopes is to ingest data from a variety of sources in the same time, transforms them, and then send them to “stash.” Elasticsearch is usually the chosen stash. (Zarzosa, 2017)

Logstash architecture uses data pipeline as the event processing engine. Its input is the collected data from variety of sources which goes through various filters (parse, process and enrich data). Logstash outputs by pushing data to a variety of destinations. Although Logstash was originally designed for log collection and parsing, its capabilities extended beyond this use case to include data enrichment and transformation features: two powerful features that are heavily used. Data enrichment, consists of adding additional information from field values. This can be done using external resources (external dictionaries with translate, geoip lookup, etc.) or resources that are shipped with Elasticsearch (urldecode, useragent parsing, key values, fingerprinting, etc.)

Logstash offers a wide range of plugins for data ingestion, data parsing, enrichment and data output. Plugins can be categorized into three types; Input plugins (data source), Filter plugins (data parsing and enrichment) and Output plugins (data destination). The plugins shipped with Logstash are enough to cover most use cases and for the rare occasions where the Logstash plugins fall short, Logstash offers the capability of adding custom plugins.

Beats are open source data shippers that are installed as agents on servers to send different types of operational data to Elasticsearch. Beats can send data directly to Elasticsearch or send it to Elasticsearch via Logstash, which can be used to parse and transform the data. For Beats some basic modules developed by the Elastic Company are: Packetbeat, Filebeat, Metricbeat, and Winlogbeat.

Kibana:

Kibana (which is the K in the ELK Stack) works as the visualization layer of the ELK Stack. It is an open-source module that queries Elasticsearch and that is meant to render the query results by means of visualizations and dashboards. Kibana core uses classics visualizations such as histograms, line graphs, pie charts, heat maps, etc. which add value to the full aggregation capabilities of Elasticsearch. (Zarzosa, 2017)

The basic features provided by Kibana are:

- Data visualization
- Browser-based interface
- Search, view, and interact with data stored in Elasticsearch indices
- Charts, tables, and maps
- Dashboards to display changes to Elasticsearch queries in real time

Kibana uses a lot of visualizations which aid the users such as: (Abdelkader Lahmadi, 2015)

- Area chart: Displays a line chart with filled areas below the lines. The areas can be displayed stacked, overlapped, or some other variations.
- Data table: Displays a table of aggregated data.
- Line chart: Displays aggregated data as lines.
- Markdown widget: Use the Markdown widget to display free-form information or instructions about dashboard.
- Metric: Displays one the result of a metric aggregation without buckets as a single large number.
- Pie chart: Displays data as a pie with different slices for each bucket or as a donut.
- Tile map: Displays a map for results of a geohash aggregation.
- Vertical bar chart: A chart with vertical bars for each bucket.

ELK stack:

The combination of Elasticsearch, Logstash and Kibana forms the ELK stack. Briefly, ELK Stack core is made of:

- Elasticsearch – Index, correlation and search engine.
- Kibana – Analytics and visualization platform.
- Logstash – Engine to collect and enrich data.
- Beats – Data shippers that collect and send data to Logstash (or directly to Elasticsearch).

The Figure below shows the components that form the Elastic Stack and the standard data flow.

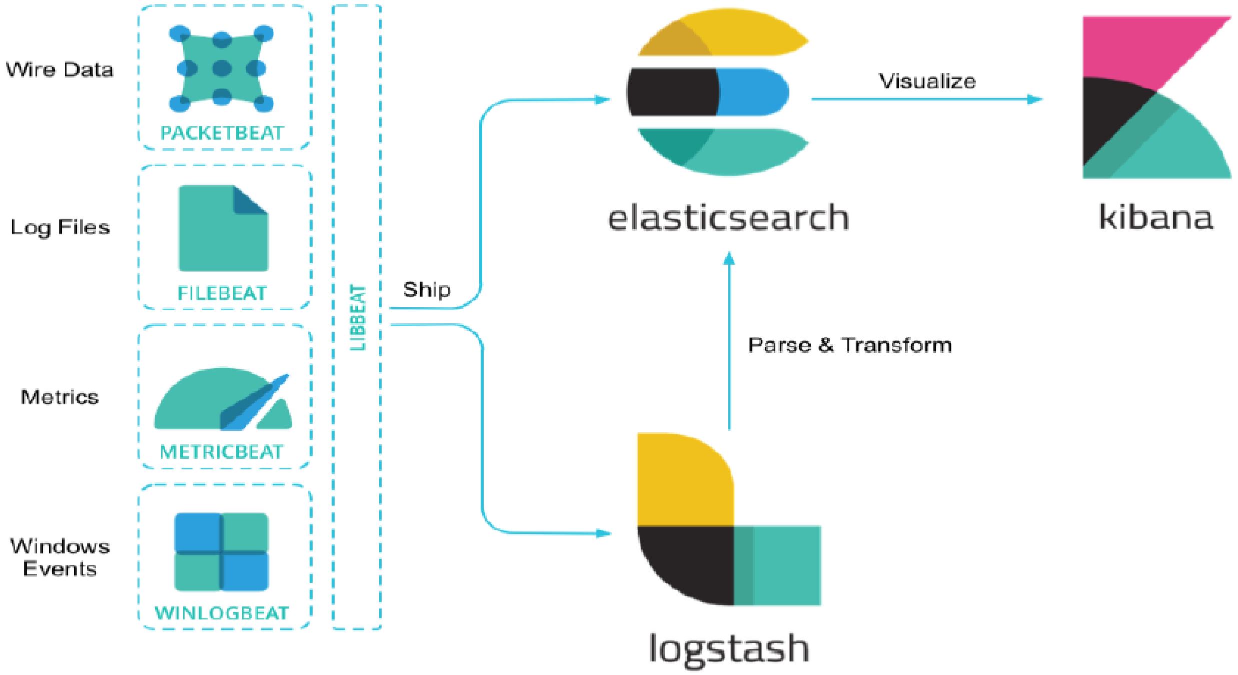


Figure 2: The standard components of ELK Stack (Zarzosa, 2017)

Technical specification of ELK stack

It is important to understand the technical specification of ELK stack to understand the potential and the limitations of the tool. Hence this section will discuss about the data sources supported, data storage capabilities, processing capabilities, event management and visualization capabilities, exposed APIs (application programming interface) and resilience of the tool.

Data sources supported

The ELK Stack is a tool capable of delivering diverse set of functionalities. These include information ingestion ability, attention on adaptability and an extensive support for wide variety of log data formats/sources. Clients can perform data indexing into elastic search either by using methods for custom software/application or by utilizing one of the ELK products. Either way, Elasticsearch would then utilize JSON configuration to store records. Since, Elastic search utilizes JSON configuration, the output of the data sources must comply to the JSON formatting.

Data storage capabilities

To fully understand data storage, it is vital to understand few core concepts of Elasticsearch (Zarzosa, 2017):

- *Near Realtime (NRT):* In Elasticsearch, slight latency exists (approximately one second) because of which it is known as a near real time search platform. This latency occurs from the time a document is indexed until the time the user can search for the document.

- *Cluster:* Cluster can be defined as a collection of n nodes (servers) which holds together the entire data. These clusters also deliver search abilities and federated indexing across all nodes.
- *Node:* A node is a single server and is a part of a cluster. Nodes are intended to store data. Other function of nodes includes participation in the cluster's indexing process and its search capabilities.
- *Index:* Index is a collection of documents. These documents resemble with each other i.e. they have similar properties e.g. an index for customer data and another index products. Name can be used to recognize an index. When performing indexing, search, update, and delete operations on the documents, names are used for referring to the index.
- *Type:* A type is a logical partition/category of an index. And the semantics of the index depends on the user. One or more types can be defined within an index.
- *Document:* Document is the basic unit of information which a user can index. Document is represented in JSON format (JavaScript Object Notation). User has the choice to store any number of documents within an index or type.
- *Shards and Replicas:* A shard in Elasticsearch is sub-part of an index. Shards provides user an option to subdivide the index into multiple pieces. Each shard is a fully-functional, independent "index" and these indexes can be present on any node in the cluster. An index can store a large amount of data. Because of this, data size can surpass the hardware limits of a single node. To overcome this issue, user can define any number of shards, when creating an index.

- *Lucene index files:* Lucene is based on the idea of inverted index. Inverted index is a versatile data structure based on computing Terms ‘frequency’ and ‘mapping’. A variety of local files in Lucene leads to the presentation of actual data indexed in the Elasticsearch.
- *Lucene*: Writing and maintaining Lucene index files are the primary responsibilities of Lucene. Elasticsearch logs metadata related to the features on top of Lucene (e.g. field mappings, index settings and other cluster metadata).
- *Retiring Data:* Data tends to become less relevant as it becomes old. To avoid irrelevant data search, Elasticsearch allows the user to easily remove the old data
- *Flat Data:* Transfer of data in the raw form and its storage in raw form is known as Flat Data.

Lucene is the component used by Elasticsearch to handle the indexing and querying on the shard level. Both Elasticsearch and Lucene write the files to the data directory. The user can increase the storage capability of an ELK cluster quite simply by adding one or many data nodes. To ensure that the data size does not exceed the node storage limit, load balancing process is triggered whenever new data nodes are added. Load balancing allows equal partitioning of data.

If the user wants to keep the retiring data around, their indices can be closed. Doing this will ensure that they still exist in the cluster, but they don’t consume any other resource apart from disk space. This is also efficient because reopening an index is much faster than restoring them from the backup. Finally, very old indices can be archived off to long-term storage, as a backup, in the case where user needs to access them in the future. Once the backup exists, the index can be permanently erased from the cluster.

Elasticsearch Curator: Elasticsearch Curator assists the user curate and manage the Elasticsearch indices and snapshots. Deleting snapshots, changing the number of replicas per shard for indices, taking a snapshot of indices, adding or removing indices from an alias, opening closed indices, forcing merging indices, changing shard routing allocation, closing indices, creating index, restoring snapshots and deleting indices are the functionalities of a Curator.

Processing capabilities of ELK

ELK performs the processes like data indexing, search, filtering, aggregation and reporting. (Zarzosa, 2017)

- *Search:* Search happens in the background. Search is done by sending parameters either through REST request body or by sending these parameters to REST request URI. In terms of the query language, ELK provides a JSON-style domain-specific one, known as Query DSL. Users can manipulate the search API.
- *Document Score:* Document score is a numeric value, a relative measure, measures the match between a document and a query. The higher the score, higher is the match between the document matches and the query. So, a lower score would represent that the document does not correspond to the query. Having said that, document score is not always necessary.
- *Filtering:* Lucene calculates a field called "document score". However, queries do not always need to compute the score. The scores are only used for filtering reasons. ELK detects these situations and only computes the scores whenever necessary. This allows for query execution optimization.
- *Aggregations:* Aggregations helps the user extract relevant statistics from the data. ELK allows users to execute aggregations and return hits as two distinct results. The types of aggregations are:

  - Pipeline: Compute statistics over the output of other aggregations and the associated metrics.
  - Matrix: Matrix, a family of aggregations whose scope is to operate on multiple fields. Doing so, they produce a matrix result based on the values extracted from the requested document fields. Scripting is not supported by this aggregation family.
  - Bucketing: A family of aggregations where each bucket has associated a document and a key. Upon execution of aggregation, all the criteria in a bucket is evaluated against every document. If the document meets the criteria, the document is considered to 'fall' in the bucket.
  - Metric: Aggregations that keep track and compute the metrics of a document collection.
- *Reporting:* ELK allows users to generate, schedule and email reports. Kibana is the usually preferred application for this purpose. The user has the option to create a report from any Kibana visualization or dashboard. Every report generated is already print-optimized, customizable and PDF-formatted. Kibana also allows users to automatically share the reports with other people.
- *Processing Scalability:* Scalability is an important feature of Elasticsearch. Elasticsearch permits rapid and fixable scaling. It can run easily on a single-node cluster or multi-node cluster. Increasing the cluster size from large cluster to a very large cluster requires additional planning and designing, but it is still rather easy.
- *Event Correlation:* Event correlation is based on handling relationships. Correlating events of different types is an essential building block of a good SIEM (Security Information and Event Management – It provides real-time analysis of security alerts). Even though Elasticsearch is not a SIEM, it does offer event correlation capability with some limitations.

ELK provides a flat view of events or entities. Elasticsearch renders four common techniques to manage relational data:

- Application-side joins: The user has the possibility to simulate a relational database by implementing joins in applications. Joins can be created by running two queries after populating two different indices (with a primary key of an index as a field in the second index). The two queries are taken from the 2 indices, i.e. first from the first index and the second from the second index. Result from the first query is used as a search term for the second query. After populating two different indices (with a primary key of an index as a field in the second index).
- Data denormalization: Denormalization is the process of improving the fetch performance by adding redundant copies. Redundant copies of data in each document eliminates the need to create joins. The user can add all the useful fields in one index, instead of Instead of having useful data in different indices. This method gives a good speed advantage.
- Nested objects: It makes sense to store closely related entities in the same document. Thus, the user has no need to create joins.
- Parent/child relationships: It is like nested objects where users can associate one entity with another. But, in nested objects, all entities live within the same document whereas in the parent-child relationship, the parent and children represent separate documents. The parent-child option allows one-to-many relationship.

A good solution would usually require a mixture of these techniques. Moreover, a lot of efforts have been carried by the community to improve the joins capabilities of Elasticsearch. For instance, there are Open Source plugins for Elasticsearch which allows fast joins across Elasticsearch indices.

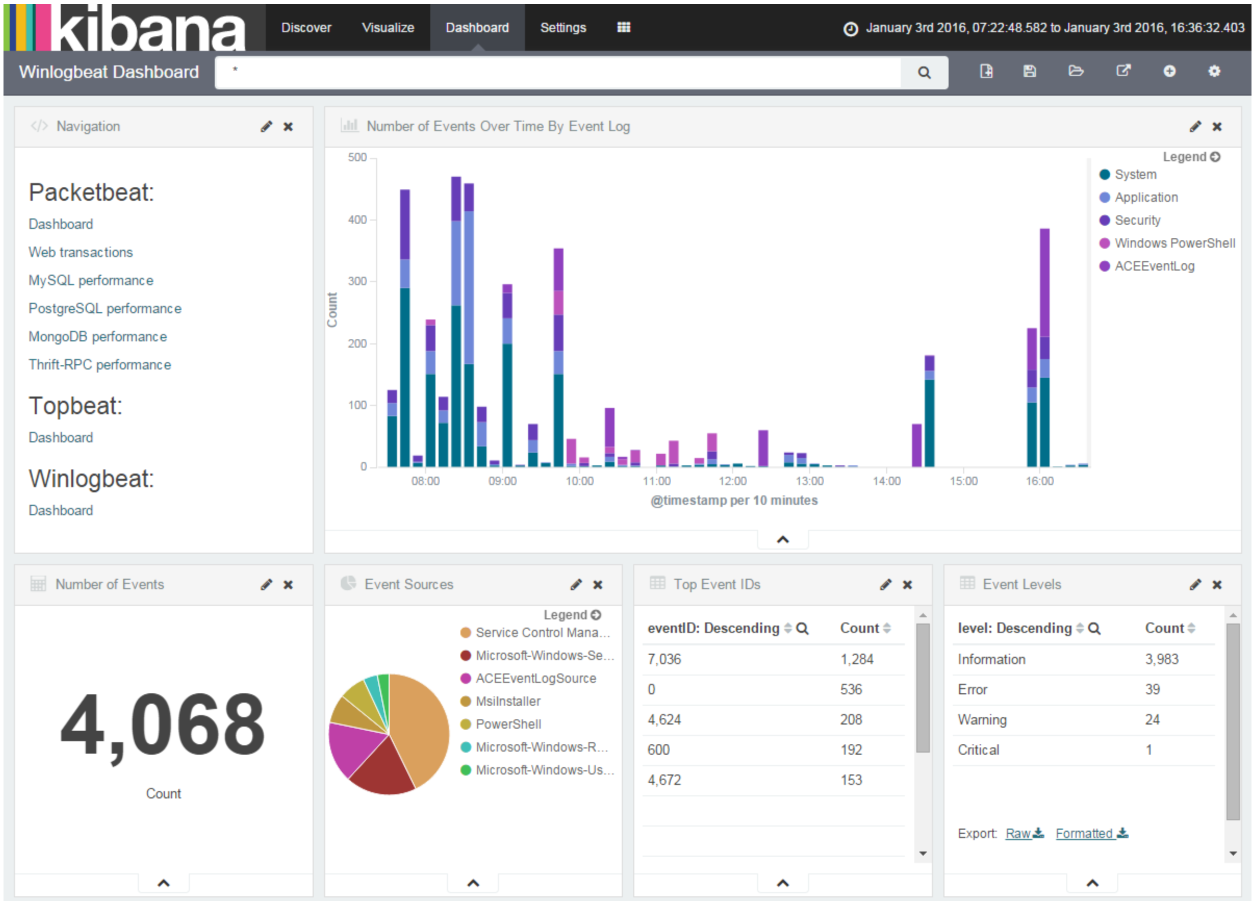
Event management and Visualization capabilities

Kibana is the visualization component of the Elastic Stack. Kibana allows users to add line/bar/scatter plots/charts/maps and to visualize large amounts of data. Primarily, Kibana allows users to create dashboards and visualizations. Kibana is also capable of data discovery and interactive exploration. This can be done using the “Discover” tab. Dashboards in Kibana can be shared and embedded as a HTML documents. Furthermore, all Kibana objects are stored as Elasticsearch documents. This feature can be useful for cloning and updating visualizations. Cloning and Updating visualization is performed using the Elasticsearch API. Kibana also offers “Dev Tools” box, useful for managing Elasticsearch indices and interacting with data using the Elasticsearch API.

- *Kibana visualizations:* Kibana visualizations are the building block of dashboards. These visualizations are based on Elasticsearch queries. A series of Elasticsearch aggregations is used to extract and process the data. Users can use the extracted data to create charts and visualize the analysis such the trends, spikes and dips. Kibana contains nine types of visualizations: Data Table, Area Chart, Line Chart, Metric, Pie Chart, Markdown Widget, Timeseries, Tile Map, and Bar Charts. Kibana also allows users to add a custom visualization. Kibana uses JavaScript to allow the users to customize visualizations.

  Advanced Kibana visualizations are based on the aggregation feature in the Elasticsearch. Different types of aggregations can be used such as bucketing, metric etc. Metric aggregations can be used to compute metrics over a set of documents, using mathematical functions and stats. The bucketing aggregations allow the user to a list of buckets, each containing a set of documents. Example of such aggregations are terms, data ranges and histograms.

Kibana Dashboards: All the types of visualization can be merged in a very customizable dashboard. The user may arrange and resize the visualizations as needed and save dashboards so to be reloaded and shared (See Figure 3).



4,068

Count

eventID: Descending	Count
7,036	1,284
0	536
4,624	208
600	192
4,672	153

level: Descending	Count
Information	3,983
Error	39
Warning	24
Critical	1

Figure 3:Sample of Kibana Dashboard (step6: loading sample kibana dashboards, n.d.)

Exposed APIs (Application Programming Interface)

APIs are a set of tools and protocols for building application software. The ELK REST APIs are exposed using JSON over HTTP and the API presents a mechanism to manipulate and access data features from a specific product. ELK API can be used to manipulate the indices. API methods can be used to manage individual indices, aliases, mappings, warmers and index settings and templates. The indexes can be created, deleted, retrieved, opened, and closed. Similar methods can be used for index mapping.

A category of API is given by the search methods based on given queries. There are more search subcategories, such as:

- Shards search.
- Suggestions, counting or validation for the search functionality.
- Search based on specific request, under the forms of query, sort, filtering, fields, preferences.
- Search based on templates.
- Aggregation searches, used more for real-time data analytics providing standard methods such as min, max, sum, average, percentiles, range or histogram.

Additional functionalities provided by the API assist in cleaning cache, optimizing and refreshing indexes or upgrading them to the latest format.

Resilience

With the growth of clusters in Elasticsearch, the possibility of network or hardware failure increases and at that point, resilience of clusters to these failures become important. A lot of effort has been invested into making Elasticsearch and Apache Lucene detect as well as cope with the increasingly difficult failures.

The failure can occur due to various factors such as network disconnections, long garbage collection, dead master, corrupted disk, timeouts, etc. In a distributed environment, failures are bound to happen. Hence, Elasticsearch offers a set of features to ensure data resiliency: (Zarzosa, 2017)

- *Snapshot/Restore API:* The snapshot and restore module allows creation of snapshots of individual indices/an entire cluster into a remote repository. These snapshots are perfect in the case of backups since they can be restored relatively quickly. But these snapshots can only be restored to versions of Elasticsearch that is able to read the index.

  Closing an old index can be a viable alternative in some cases: indices get old, they reach a point where they are almost never accessed. These indices can be closed. For some cases, this is a better option because reopening an index is much quicker than restoring it from backup. ELK has certain limitations: at any given point of time, only one snapshot process can be executed within a given cluster. Also, while the snapshot is being created, the shard cannot be moved to a different node. The downside of not being able to move a shard is that, it can interfere with the processes such allocation filtering and load balancing.
- *Sequence Numbers:* Sequence Number is a number assigned by the to the operations that occur on primary shards. One of the operations is speeding up the replica recovery process when a node is restarted. Sequence numbers will allow us to copy over only the data which

has really changed. Previously, every segment was copied since changed data was not being tracked.

- *Multiple Data Paths:* It is possible to set multiple data paths when configuring Elasticsearch. The goal is to spread data across multiple disks or locations, thus, limiting disk failures impacts.
- *Checksums:* Both Elasticsearch and Lucene perform checksums on different stages of data to ensure that the data is not corrupt. To make sure that the created snapshot does not contain corrupted files, the snapshot process verifies the checksums for each file which is a part of the snapshot
- *Elasticsearch Cluster:* A node is a running instance of Elasticsearch and a cluster is a group of nodes. Nodes with the same cluster name work together to share data and provide failover and scale. Even a single node can form a cluster.

Elasticsearch can scale out to hundreds (or even thousands) of servers and is capable of handling petabytes of data. The operations happening automatically in the background are:

- Partitioning the documents into different containers or shards. They can be stored on a single node or on multiple nodes.
- Balancing these shards across the nodes into the cluster to spread the indexing and search load.
- Duplicating each shard to provide redundant copies of data but also to prevent data loss in case of hardware failure.
- Routing requests from any node in the cluster to the nodes that hold the data the user is interested in. Seamlessly integrating new nodes as the cluster grows or redistributing shards to recover from node loss.

Implementation and configuring ELK stack

After developing a basic understanding about ELK stack and its capabilities, this chapter will now talk about the basic implementation and configuration of ELK stack. Before discussing about implementation, it is important to understand ELK stack architecture which is mentioned in Figure 4.

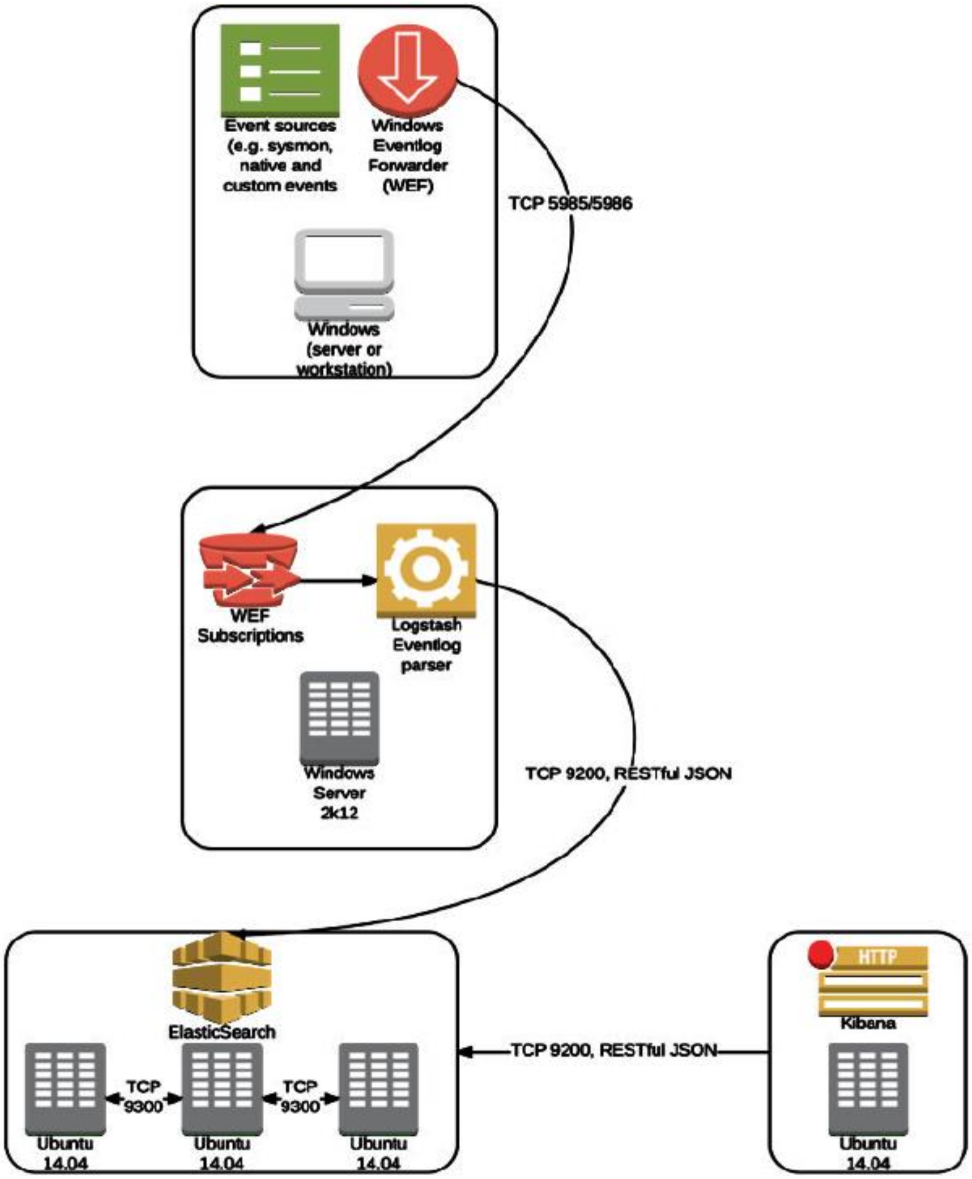


Figure 4: Architecture Overview (LEWIS, 2015)

Deployment of ELK

To take full advantage of ELK capabilities, the user needs to use following arrangement that resembles deployment where the user is making use of 4 tiers:

- The input tier: It consists of data consumed from source and consists of Logstash instances with adequate input plugins.
- The message broker: An entity which serves as a buffer which holds ingested data. It also works as a failover protection.
- The filter tier: It applies data parsing.
- The indexing tier: It moves the processed data to Elasticsearch.

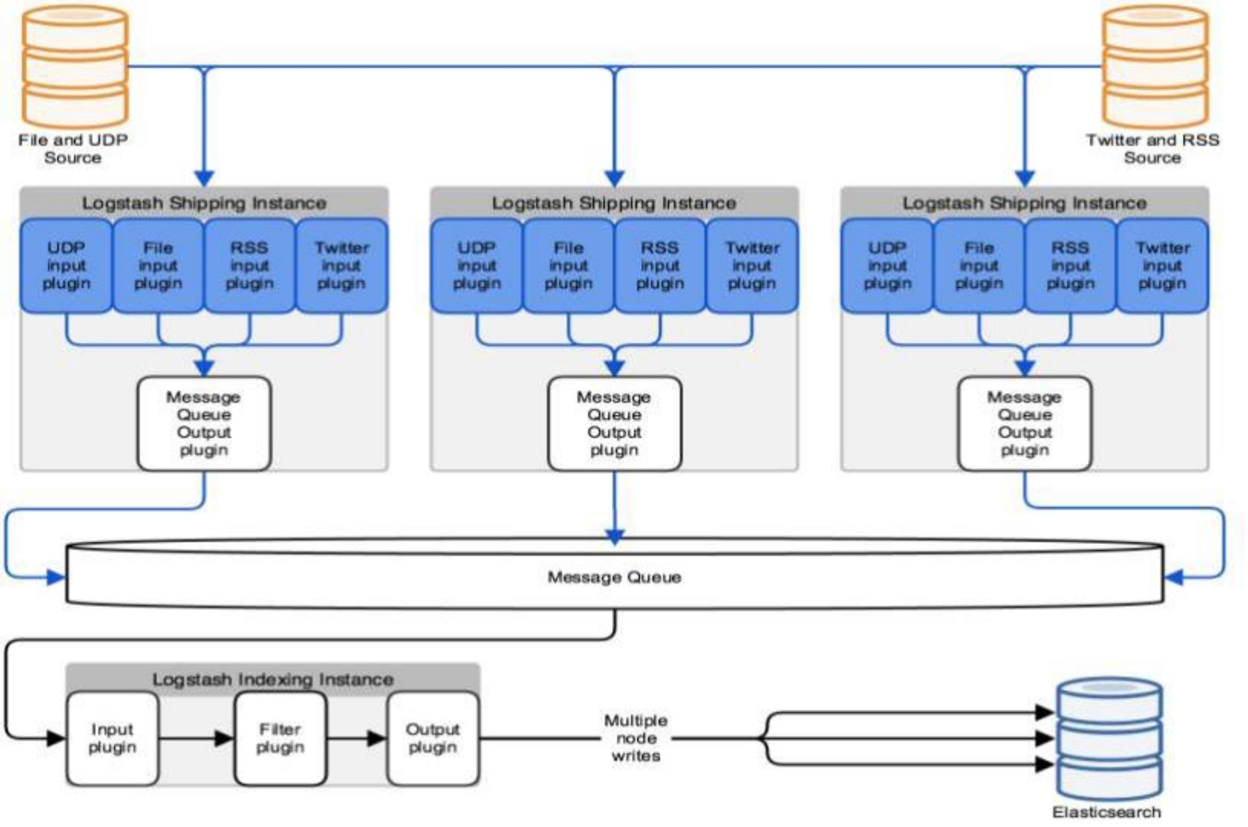


Figure 5: ELK set-up (Zarzosa, 2017)

Elastic Cloud

Elastic Cloud is a hosted Elasticsearch solution. Not only, it has the capabilities of the Elasticsearch and Kibana, but also reduces deployment time without the worry of hosting and infrastructure maintenance. Secure cluster running on Amazon can be easily deployed and managed by anyone. Elastic Cloud runs with the latest versions of Elasticsearch and Kibana.

Advantages of Elastic Cloud:

- Easy Scaling and upgrading. Necessity being the driving factor behind the change in structure.
- Free instance of Kibana, and cluster-backup 30 minutes.
- 24/7 hardware and infrastructure monitoring by the Elastic team.
- Elastic Cloud supports most of the security features that are part of the Elastic Stack. These features are designed to:
    - Prevent unauthorized access with password protection, and role-based access control.
    - Preserve the integrity of data with message authentication and SSL/TLS encryption.

SYSMON

To truly prepare the environment, several areas of logging should be considered, especially for Windows hosts. In the most basic form, to detect human adversaries on an organization's network, additional auditing can be performed for windows hosts. This would reveal the necessary records to detect human adversaries.

Advantages of SYSMON (Per Microsoft, Sysmon provides the following capabilities):

- Logs process creation with full command line for both current and parent processes.
- Records the hash of process image files using SHA1 (the default), MD5 or SHA256.
- Includes a process GUID in process create events to allow for correlation of events even when Windows reuses process IDs.
- Optionally logs network connections, including each connection's source process, IP addresses, port numbers, hostnames and port names.
- Detects changes in file creation time to understand when a file was really created.
- Modification of file create timestamps is a technique commonly used by malware to cover its tracks.
- Generates events from early in the boot process to capture activity made by even sophisticated kernel-mode malware.

SYSMON SETUP (WITHOUT GPO):

Run the following command in command prompt as admin:

- Sysmon.exe –i –h SHA256 –n

LOGSTASH

There are some issues with logstash which occur during the setup of logstash. Firstly, logstash would not pull data from the forwarded event log file. The subscriptions can be configured to save data to another file. Secondly, logstash has a memory leak and a heavy load would prevent it from running. To solve these issues, slight tweak is needed in the architecture. NX Log CE needs to be installed on the Windows Server event log collector and should be used to forward data to a logstash instance running on an Ubuntu instance. This logstash instance would listen for connections' data from NX log and forward to the ElasticSearch cluster.

INSTALL NX LOG:

Download and install binary. Create a configuration file and place it in the configuration directory.

LOGSTASH CONFIGURATION FILE SETUP:

- Install java
  - apt-get install default-jre
  - update-alternatives --config java

    Capture location (e.g. /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java)
  - Vi /etc/environment

    Add new line

    JAVA_HOME=”/usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java”

    source /etc/environment

    echo $JAVA_HOME
- Create a configuration file “logstash.conf” and place in the /usr/share/logstash/bin/ folder.

RUNNING LOGSTASH;

- C:\logstash\bin\logstash.bat agent –f logstash.conf

ELASTICSEARCH

Advantages of Elasticsearch:

- Scalable horizontal compute and storage (similar to Hadoop).
- ElasticSearch can accommodate node failure and automatically re-distribute shards without data loss.
- ElasticSearch leverages parallel processing to execute queries on massive data sets very quickly.

- Open source (free).
- Great for long tail analysis and querying.

ELASTICSEARCH SETUP:

- Prior to starting ElasticSearch, open the elasticsearch.yml file,
  - Remove the “#” on the line that contains “#cluster.name: elasticsearch”.
  - Change the “elasticsearch” to a different name that describes the cluster.
  - Note that new ElasticSearch instances will auto-join the cluster if they have the same cluster name.
  - Note: setup iptables or put the ElasticSearch cluster behind a firewall. ElasticSearch does not authenticate any connections by default. ElasticSearch listens on port 9200 for RESTful HTTP connections.
- Install java
  - apt-get install default-jre
  - update-alternatives --config java

    Capture location (e.g. /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java)
  - Vi /etc/environment

    Add new line

    JAVA_HOME=”/usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java”

    source /etc/environment

    echo $JAVA_HOME

START ELASTICSEARCH:

- ./elasticsearch

KIBANA

Advantages of Kibana:

- Allows user to easily query Elasticsearch by providing a beautiful interface.
- Allows quick filtering to perform long tail analysis and anomaly detection.

KIBANA SETUP:

- Edit the config.js file and point Kibana to the Elasticsearch cluster
  - Change the IP address to the host IP address on line “elastic search: http://127.0.0.1:9200”
- vi /etc/init/kibana.conf
  - # auto start kibana
  - start on filesystem
  - exec /usr/share/kibana/bin/kibana
- chmod -R 755 /usr/share/kibana/
- ln -s /etc/init/kibana.conf /etc/init.d/kibanae
- service kibana start

Creating New Connectors

Creating Logstash Plugins: The normal users can create their own Logstash plugin in a matter of seconds. Logstash plugins can perform functions such as insert, delete, update, query and enrich. It provides correct directory structure, gemspec files, and dependencies so that the user can start adding custom code to process the data with Logstash. For example, ‘generate’ subcommand of bin/logstash-plugin can be used as the foundation to create a new Logstash plugin with templatized files.

Creating a Beat: Beats are a collection of lightweight daemons. These daemons are capable of collecting operational data from the user servers and transferring them to the Elasticsearch/Logstash. Library libbeat contains the common parts for all the beats. The library also contains the packages for transferring data to Elasticsearch/Logstash, for configuration file handling, logging, signal handling, etc.

All Beats are written in Go, that’s why it is harder to configure. Very largely said, a simple Beat has two principle components:

- Publisher: Sends the data to the specified output, such as Elasticsearch or Logstash.
- Collector: Collects the actual data.

Elasticsearch Clients: One of the important aspect of Elasticsearch is that, it is independent of any programming language. All the APIs for indexing, searching and monitoring can be accessed using HTTP and JSON. This enables easy integration with any language which has HTTP and JSON capabilities. Nevertheless, Java (the language Elasticsearch and Lucene are implemented in) is very dominant. Users can manipulate the Java client to perform standard index, get, delete and search operations on an existing cluster. Users can also manipulate the client to make administrative tasks on a running cluster. Other clients like Python, Perl, Ruby, PHP, .NET and Groovy, can be found at the official Elastic website.

PowerShell Logging

Enabling enhanced PowerShell logging is very important in cyber security perspective both for incidence response and forensics and it is a lesser known fact that it is actually easy to ship its logs to an ELK stack for analysis. Enhanced PowerShell logging, means enabling Module & Script Block Logging. *Module logging* allows to specify the PowerShell *modules* that one want to *log*. *Script block* stores PowerShell commands that are run without the output. Those two enhancements started with Windows Management Framework (WMF) version 4.0 and 5.0.

- Module Logging: Module logging records execution details as PowerShell executes, including variable initialization and command invocations. Module logging will record portions of scripts, some de-obfuscated code, and some data formatted for output. This logging will capture some details missed by other PowerShell logging sources, though it may not reliably capture the commands executed. Module logging has been available since PowerShell 3.0. Module logging events are written to Event ID (EID) 4103. (Dunwoody, 2016)
- Script Block Logging: PowerShell v5 and KB 3000850 introduced deep script block logging. After, enabling script block logging, PowerShell records the content of all script blocks that it processes. If a script block uses dynamic code generation, PowerShell will log the invocation of this generated script block as well. This provides complete insight into the script-based activity on a system – including scripts or applications that leverage dynamic code generation to evade detection. Script Block logging events are written to Event ID (EID) 4104. (Dunwoody, 2016)

Setup of PowerShell Logging in ELK stack

This section will explain the basic requirements and configurations to be done on all the systems to enable enhanced PowerShell logging. Enhanced PowerShell logging will be required in the test cases mentioned in the next section to get logs of processes created using PowerShell command line.

Requirements

For enhanced PowerShell logging using ELK stack the requirements are:

- Windows Management Framework (WMF) 5.0 or 5.1 (Download)
- Sysmon Installed (Version 6.X or above)
- Winlogbeat forwarding logs to an ELK Server
- PowerShell 5.0 or above
- Logstash Filter Plugins like KV, GROK & MUTATE

Installation of Windows Management Framework (WMF)

Windows 10 does not require any software updates to support enhanced PowerShell logging. But, for Windows 7/8.1/2008/2012, upgrading PowerShell to enable enhanced logging in PowerShell 5.0 requires:

- .NET framework 4.5
- Windows Management Framework (WMF) 4.0 (Windows 7/2008 only)
- Windows Management Framework (WMF) 5.0

- Windows 7 and 2008 R2 must be upgraded to Windows Management Framework (WMF) 4.0 prior to installing WMF 5.0.
- The appropriate WMF 4.0 update according to operating systems,
    - Windows 8.1/Server 2012 R2 – KB3000850
    - Server 2012 – KB3119938
    - Windows 7/Server 2008 R2 SP1 – KB3109118

Downloading these updates from Microsoft may require the completion of an automated request process.

To install WMF 5.0 from Windows Explorer (or File Explorer):

- Navigate to the folder into which the MSU file is downloaded.
- Double-click the MSU to run it.

To install WMF 5.0 from Command Prompt:

- After downloading the correct package for the computer's architecture, open a Command Prompt window with elevated user rights (Run as Administrator). On the Server Core installation options of Windows Server 2012 R2 or Windows Server 2012 or Windows Server 2008 R2 SP1, Command Prompt opens with elevated user rights by default.
- Change directories to the folder into which the downloaded or copied WMF installation package is placed.
- Run the following command: “run (name of the msu file).msu /quiet”

Setting up Enhanced PowerShell Logging

Using Group Policy Object Settings

If there is a domain controller set up in the environment with Active Directory services enabled, it can be used to create Audit Policies and apply them to the entire whole domain. Below are the steps to be followed to setup enhanced PowerShell logging using Active Directory:

- Create and name new Group Policy Object (GPO)
- Edit the newly created GPO
- Browse to "Windows PowerShell" Settings
    - In Group Policy Management Editor, browse to Computer configuration > Administrative Templates: Policy Definitions > Windows Components > Windows PowerShell
- Turn on Module Logging
    - Right click on "Turn on Module Logging", select Edit and check the "Enabled" option
    - Once enabled, the "Show" options next to "Modules Names" will be available.
    - Click on "Show" and a Show Content window will pop up
    - Click on value and add an "*" in order to log all PowerShell modules
    - Click OK on the "Show Content Window" to exit out of it
    - Click Apply and OK
- Turn on Script Block Logging
    - Right click on "Turn on PowerShell Script Block Logging" and select Edit
    - Check the "Enabled" option

    - Check "log script invocation start/stop" options
    - Click on Apply and OK
- Link new existing GPO
    - Go back to Group Policy Management and right click on the Domain name
    - Select Link an Existing GPO
    - Select the GPO created and click OK
    - Force Group Policy updates on the machines connected on the Domain by using the command “gpupdate /force” in command prompt.
- Testing Enhanced PowerShell
    - Run a simple PowerShell Command
    - Check events in Event Viewer and a 4103-event showing the module that was used in the command should show up

Using Local Policy Settings

When enhanced PowerShell logging must be enabled on individual machines, i.e., when the machines are not connected to a domain controller, local policies have to be created on every machine either on either gpedit or regedit. It can be achieved by following the above mention steps on a local policy editor.

Ship PowerShell logs to ELK

So, after testing that the enhanced PowerShell logging is working fine, next step will be shipping the logs to an ELK Stack. Considering the fact, that there is already a working ELK stack and all the machines have a log shipper, next step will be adding configuration to the “Winlogbeat” configuration file. The steps to ship PowerShell logs to ELK stack are:

- Open "Winlogbeat" configuration as Administrator with Notepad and add the following under Winlogbeat.event_logs section:
    - name: Microsoft-Windows-PowerShell/Operational

      event_id: 4103, 4104
- Save the file for changes and restart Winlogbeat service.

Creating a Logstash Filter

The reason for creation of an extra Logstash Filter here is, that the data being shipped to ELK of the EID 4103, is split in two fields, [event_data][Payload] and [event_data][ContextInfo]. In this case, [event_data][Payload] should give modules information, but Payload has everything as a long string which is then stored as a long string in Elasticsearch without creating extra fields. Hence, adding a new configuration file in conf.d folder of Logstash will help us solve this problem. Adding the following script to the configuration file will create the required Logstash filter. After writing the script in the configuration file, the next step will be saving the “.conf” file and restarting the Logstash service. Next step, is testing the Logstash logs to make sure everything runs smoothly.

Understanding PowerShell logs on ELK

After the setup for enhanced PowerShell logging is complete, Kibana starts showing extra fields that can be added as columns when visualizing 4103 logs. Basically, now using the functionalities of Kibana, understanding the details of an event happening through PowerShell becomes easier, for both incidence response and forensics. (Note: 4103 and 4014 are the events associated with PowerShell activity.)

Due to its property of going undetected, usage of PowerShell by attackers to perform attacks has increased over the past few years. It’s a powerful environment that can accomplish tasks in every phase of an attack from injecting shellcode into memory to post exploitation situational awareness and lateral movement. (Bandos, 2017) Hence PowerShell logging becomes a vital subject to understand and detect lateral movement.

Detecting Lateral Movement

This section is a description of the test cases performed to study and analyze how various tools can be used to perform lateral movement. The analysis of the logs generated in each process helped in documenting the proof of execution which can later work as a baseline to develop filters in ELK stack to create a detection mechanism against these specific tools.

The 24 commonly used tools to perform lateral movement which will be discussed in the test cases are: PsExec, PowerShell, WinRM, WinRS, wmic, wmiexec.vbs, PWDumpX, Quarks PwDump, WCE, WCE (Remote Login), Golden Ticket (mimikatz), Silver Ticket (mimikatz), AT, RDP, mimikatz, MS14-058 Exploit, ntdsutil, vssadmin, net user, Find-GPOPasswords.ps1, csvde, ldifde, timestomp and wevtutil. These test cases explain the basic information about the tool, how it is used and states the way to detect that particular tool's usage.

Every test case will follow the following structure:

- Basic Information: This consists of the tool name, tool description and how the tool is being used in performing the attack.
- Operating condition: This explains the conditions required for the tools to execute successfully, such as, what kind of user privilege is required, what kind of protocol is used to communicate and what service are required.
- Information acquired from log: This consists the documentation of information acquired from logs generated when the tool was tested.
- Log generation: This section shows the logs that had to be monitored to analyze the event.
- Execution confirming evidence: This displays the proof of execution and is the result of the study.

- Remarks, if any

Test case 1: Using Psexec

Basic Information

- Tool Name: PsExec
- Category: Command Execution
- Tool Overview: Executes a process on a remote system
- Tool Usage During an Attack:
    - Used to remotely execute a command on client and servers in a domain.

      Source host: PsExec command execution source.

      Destination host: The destination logged in by the PsExec command.

Operating Condition

- User Account Type:
    - Source host: Standard user
    - Destination host: Administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 135/tcp, 445/tcp, a random high port (When executing in a domain environment, communication for Kerberos authentication with the domain controller occurs.)
- Service: Not Applicable

Information Acquired from Log

- Source host: A registry to the effect that the PsExec License Agreement has been entered is registered.
- Destination host: The fact that the "PSEXESVC" service has been installed, started, and ended is recorded.
- Execution history (Sysmon/audit policy)
    - Source host: The fact that the PsExec process was executed and that connection was made to the destination via the network, as well as the command name and argument for a remotely executed command are recorded.
    - Destination host: The fact that the PSEXESVC's binary was created and accessed, and that connection was made from the source via the network, as well as the command name and argument for a remotely executed command are recorded.

Log Generation

- Source Host

  Security Event Log
    - Event ID: 4688 (A new process has been created)

      4689 (A process has exited)
    - Process Information: -Process Name: "[Execution File (psexec.exe)]"
- Destination Host

  Security Event Log
    - Event ID: 7045 (A service was installed in the system)

7036 (The service state has changed) The "PSEXESVC" service enters the "Executing" state before executing a remote process and enters the "Stopped" state after the execution.

5156 (The Windows Filtering Platform has allowed a connection) Communication occurs from the source host to the destination with a random high port (1024 and higher) as the destination port.

5140 (A network share object was accessed)

4672 (Special privileges assigned to new logon) Before this event occurs, the event 4624 occurs.

4656 (A handle to an object was requested)

4663 (An attempt was made to access an object)

5140 (A network share object was accessed)

5145 (A network share object was checked to see whether client can be granted desired access) The Event ID is recorded several times.

4656 (A handle to an object was requested)

4660 (An object was deleted)

4658 (The handle to an object was closed)

- Process Information: -Process Name: "PSEXESVC"

Execution confirming evidence:

- Confirmation of PsExec was execution:
  - Source host: Event ID 4689 (A process has exited) of psexec.exe was recorded in the event log with the execution result (return value) of "0x0".

  - Destination host: PSEXESVC.exe is installed.

Remarks

Additional Event Logs That Can Be Output:

Information related to the process execution using PsExec may be recorded to the Destination host if enhanced PowerShell logging is switched on.

Test Case 2: PowerShell (Remote Command Execution)

Basic Information

- Tool Name: PowerShell (Remote Command Execution)
- Category: Command execution
- Tool Overview: A command line tool that can be used for Windows management and settings. In addition to the host that executes PowerShell, this tool enables commands to be executed on other hosts via a network.
- Tool Usage During an Attack :
  - Used to change settings to enable the Domain Controller and other hosts on the network to perform operations requiring administrator rights.

    Source host: PowerShell command execution source.

    Destination host: The destination logged in by the PowerShell command.

Operating Condition

- User Account Type: PowerShell can be used by standard users. To execute a script for changing settings, appropriate rights are needed on the host to change settings.
- Targeted OS: Windows

- Domain: Not required
- Communication Protocol: Not required to manage within local machines. To manage other hosts, use 80/tcp or 5985/tcp for HTTP and 443/tcp or 5986/tcp for HTTPS.
- Service: Windows Remote Management (WS-Management)

Information Acquired from log

- Execution history (Prefetch).
- Execution history (Sysmon, audit policy). The end event of PowerShell allows execution results to be confirmed. With audit policy, it is possible to confirm the occurence of communication from the source host to the destination host 5985/tcp (HTTP) or 5986/tcp (HTTPS).

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    5156 (The Windows Filtering Platform has allowed a connection)
  - Process Information: -Process Name: "C:\Windows\System32\Windows PowerShell\v1.0\powershell.exe "
- Destination Host

  Security Event Log

  - Event ID: 5156 (The Windows Filtering Platform has allowed a connection)

4624 (An account was successfully logged on) The date immediately after the wsmprovhost.exe process was created (Event ID 4688) and before it ended (Event ID 4689).

4634 (An account was logged off)

Execution confirming evidence:

- If the following logs that have the same log time are found, it confirms that a remote command was executed. This also applies to Prefetch.
  - Source host: Event ID 4689 (A process has exited) of PowerShell was recorded in the event log with the execution result (return value) of "0x0".
  - Destination host: Event ID 4689 (A process has exited) of wsmprovhost.exe was recorded in the event log with the execution result (return value) of "0x0".

Remarks

Additional Event Logs That Can Be Output:

Depending on the command that is executed, logs output by the command may be recorded at the destination host.

Test Case 3: WinRM

Basic Information

- Tool Name: WinRM
- Category: Command execution
- Tool Overview: Executes a command on a remote host
- Tool Usage During an Attack :

- Used for carrying out an investigation before executing a remote command.

  Source host: WinRM command execution source.

  Destination host: The machine accessed by the WinRM command.

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 5985/tcp (HTTP) or 5986/tcp (HTTPS)
- Service: Destination host: Windows Remote Management (WS-Management)

Information Acquired from log

- Execution history (Prefetch).
- Source host: Execution history (Sysmon / audit policy).
- Destination host: Connection from the source host.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    5156 (The Windows Filtering Platform has allowed a connection)

    1 (Process Create)

    5 (Process Terminated)

    - Execution History: Prefetch - File name: C:\Windows\Prefetch\CSCRIPT.EXE-D1EF4768.pf, Confirmable Information (the following can be confirmed using this tool: WinPrefetchView).

- Destination Host

  Security Event Log

    - Event ID: 5156 (The Windows Filtering Platform has allowed a connection)
      4624 (An account was successfully logged on)
      4656 (A handle to an object was requested)
      4658 (The handle to an object was closed)
      5156 (The Windows Filtering Platform has allowed a connection)
      4769 (A Kerberos service ticket was requested)

Execution confirming evidence:

- Source host: A log indicating that cscript.exe accessed the destination host with Event IDs 1 and 5 of the event log.

Test Case 4: WinRS

Basic Information

- Tool Name: WinRS
- Category: Command execution
- Tool Overview: Executes a command on a remote host
- Tool Usage During an Attack:
    - This tool is sent by the BITS and remotely executed using winrs.

Source host: WinRS command execution source.

Destination host: The machine accessed by the WinRS command.

Operating Condition

- User Account Type:
  - Source host: Standard user
  - Destination host: Administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 5985/tcp (HTTP) or 5986/tcp (HTTPS)
- Service: Destination Host: Windows Remote Management (WS-Management)

Information Acquired from log

- WinRM execution log.
- Execution history (Prefetch).
- Execution history (Sysmon, audit policy).
- Recording of communication via Windows Filtering Platform.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4648 (A logon was attempted using explicit credentials)

    5156 (The Windows Filtering Platform has allowed a connection)

- Application and Service: Microsoft\Windows\Windows Remote Management\Operational. That fact that WinRS was executed is recorded.

- Destination Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    5156 (The Windows Filtering Platform has allowed a connection)
  - Application and Service: Microsoft\Windows\Windows Remote Management\Operational. The fact that the WinRS process corresponding to the log at the source host was executed is recorded.

Execution confirming evidence:

- The execution of WinRS is recorded in the event log "Application and Service\Microsoft\Windows\Windows Remote Management\Operational".

Remarks

Additional Event Logs That Can Be Output:

- Logs by a command execution via WinRS may be recorded.

Test Case 5: WMIC (Windows Management Instrumentation Command Line)

Basic Information

- Tool Name: WMIC (Windows Management Instrumentation Command Line)
- Category: Command execution
- Tool Overview: A tool used for Windows system management
- Tool Usage During an Attack:
    - Used to acquire information on the remote system or to execute a command with WMI.

      Source host: wmic command execution source.

      Destination host: The host accessed by the wmic command.

Operating Condition

- User Account Type: Standard user but depending on the command executed on the remote side, administrator privileges may be required.
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 135/tcp, 445/tcp, a randomly selected TCP port 1024 or higher
- Service: Windows Management Instrumentation, Remote Procedure Call (RPC)

Information Acquired from log

- Execution history (Prefetch).
- Process execution details (the argument to wmic) and execution success or failure (the return value) (Sysmon and audit policy).

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)
  - Process Information: - Process Name: "C:\Windows\System32\wbem\WMIC.exe"
- Destination Host

  Security Event Log

  - Event ID: 1 (Process Create)

    5 (Process Terminated)

Execution confirming evidence:

- If the following logs that have the same log time are found at source host and destination host, a remote connection was made:
  - Source host:Event ID 4689 (A process has exited) of WMIC.exe was recorded in the event log with the execution result (return value) of "0x0".
  - Destination host: It is recorded in the event log that WmiPrvSE.exe was executed with the event IDs 1 and 5.

Remarks

Additional Event Logs That Can Be Output:

Depending on the process called by wmic, the process-specific logs may be recorded. If the user exists on the Active Directory, the authentication request may be recorded in the Domain Controller. However, it is not possible to determine whether such an authentication request was made by wmic or others.

Test Case 6: wmiexec.vbs

Basic Information

- Tool Name: wmiexec.vbs
- Category: Command Execution
- Tool Overview: A tool used for Windows system management
- Tool Usage During an Attack:
    - Executes a script for other hosts.

      Source host: The source that executes wmiexec.vbs.

      Destination host: The machine accessed by the wmiexec.vbs.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 135/tcp, 445/tcp
- Service: Not Applicable

Information Acquired from Log

- Execution history (Prefetch).
- Execution history (Sysmon/audit policy).
- File creation/delete history (Audit policy).
- Execution history (Sysmon).

Log Generation

- Source Host

Security Event Log

- Event ID: 4688 (A new process has been created)

  4689 (A process has exited)

  5156 (The Windows Filtering Platform has allowed a connection)

- Destination Host

  Security Event Log

  - Event ID: 4656 (A handle to an object was requested)

    4663 (An attempt was made to access an object)

    4658 (The handle to an object was closed)

    5142 (A network share object was added)

    5145 (A network share object was checked to see whether client can be granted desired access)

    4656 (A handle to an object was requested)

    4660 (An object was deleted)

    4658 (The handle to an object was closed)

    5144 (A network share object was deleted.)

  - Share Name: Share Information -> Share Name ("\\*\WMI_SHARE")

Execution confirming evidence:

- Destination host: The "WMI_SHARE" share has been created and deleted.

Test Case 7: PWDumpX

Basic Information

- Tool Name: PWDumpX
- Category: Password and Hash Dump
- Tool Overview: Acquires a password hash from a remote host.
- Tool Usage During an Attack :
  - Uses the acquired hash to perform attacks such as pass-the-hash.

    Source host: PWDumpX execution source.

    Destination host: The destination logged in by PWDumpX.

Operating Condition

- User Account Type: Source host: Standard user

  Destination host: Administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 135/tcp, 445/tcp
- Service: Not Applicable

Information Acquired from log

- Both hosts: Execution history (Prefetch)
- Destination host: The fact that the PWDumpX service has been installed and executed is recorded.
- The fact that the PWDumpX service has been sent from the source host to the destination host and then executed is recorded.

- The fact that a text file is used to create and receive hash information is recorded.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4663 (An attempt was made to access an object), Data is written to the above file multiple times
  - Object: -Object Name: "[Path to Tool]\[Destination Address]-PWHashes.txt.Obfuscated"
- Destination Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    5145 (A network share object was checked to see whether client can be granted desired access)

    4663 (An attempt was made to access an object)

    7045 (A service was installed in the system)

Execution confirming evidence:

- Source host: If "[Path to Tool]\[Destination Address]-PWHashes.txt" has been created, it was successfully executed.

Test Case 8: Quarks PwDump

Basic Information

- Tool Name: Quarks PwDump
- Category: Password and Hash Dump
- Tool Overview: Acquires the NTLM hash of a local domain account and cached domain password. Information in a machine including NTDS.DIT files can be specified and analyzed.
- Tool Usage During an Attack:
  - Uses acquired hash information to perform logon authentication on other hosts.

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: Not Applicable
- Service: Not Applicable

Information Acquired from log

- Source host: Execution history (Prefetch).
- Execution history (Sysmon/audit policy).
- A record that the temporary file ("SAM-[Random Number].dmp") has been created.

Log Generation

- Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4656 (A handle to an object was requested)

    4663 (An attempt was made to access an object)

    4658 (The handle to an object was closed)

    4660 (An object was deleted)
  - Object: -Object Name: ("C:\Users\[User Name]\AppData\Local\Temp\SAM-[Random Number].dmp")

Execution confirming evidence:

- A temporary file ("SAM-[Random Number].dmp") was created and deleted.

Test Case 9: WCE

Basic Information

- Tool Name: WCE (Windows Credentials Editor)
- Category: Password and Hash Dump
- Tool Overview: Acquires password hash information in the memory of a logged in host.
- Tool Usage During an Attack :
  - Uses the acquired hash information to perform pass-the-hash and other attacks.

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows
- Domain: Not required

- Communication Protocol: Not Applicable
- Service: Not Applicable

Information Acquired from log

- Execution history (Prefetch).
- The fact that a tool was executed, and the option used during tool execution (Sysmon).
    - Reference of lsass.exe by the tool (Sysmon).
    - Creation / deletion of a file (audit policy).

Log Generation

- Host

  Security Event Log

    - Event ID: 4688 (A new process has been created)

      4689 (A process has exited)

      4656 (A handle to an object was requested)

      4663 (An attempt was made to access an object)

      4658 (The handle to an object was closed)

      4660 (An attempt was deleted)

      1 (Process Create)

      5 (Process terminated)
    - Object: -Object Name: ("C:\Users\[User Name]\AppData\Local\Temp\wceaux.dll"

Execution confirming evidence:

- The "C:\Users\[User Name]\AppData\Local\Temp\wceaux.dll" file was created and deleted.

Test Case 10: WCE

Basic Information

- Tool Name: WCE (Remote Login)
- Category: pass-the-hash, pass-the-ticket
- Tool Overview: Executes a command with higher privileges using the hash of the acquired password.
- Tool Usage During an Attack:
    - Remotely executes a command on another machine by using a password hash for a user with Administrator privileges which belongs to Active Directory

      Source host: WCE execution source.

      Destination host: The destination logged in by WCE.

Operating Condition

- User Account Type: Local administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: A random 5-digit port (WMIC)
- Service: Not Applicable

Information Acquired from log

- Source host: Execution history (Prefetch).
- A record of the fact that WCESERVICE was installed and executed.
- Both source host and destination host: WMI execution history and Windows Filtering Platform log.

- Destination host: Login has occurred remotely.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4656 (A handle to an object was requested)

    4663 (An attempt was made to access an object)

    4658 (The handle to an object was closed)

    5156 (The Windows Filtering Platform has allowed a connection)

    7045 (A service was installed in the system)

    7036

    1 (Process Create)

    5 (Process terminated)

    8 (CreateRemoteThread detected)

  - Service Name: "WCESERVICE"

- Destination Host

  Security Event Log

  - Event ID: 5156 (The Windows Filtering Platform has allowed a connection)

    4624 (An account was successfully logged on)

    4634 (An account was logged off)

    4688 (A new process has been created)

    4689 (A process has exited)

    9 (RawAccessRead detected)

Execution confirming evidence:

- Source host: The fact that WCESERVICE was installed and executed is recorded.
- Destination host: The fact that a logon was made from a remote host is recorded.
- Both source host and destination host: The fact that communication using WMI occurred is recorded.

Test Case 11: Golden Ticket (mimikatz)

Basic Information

- Tool Name: Mimikatz (Golden Ticket)
- Category: Capturing the Domain Administrator Privilege and Account Credentials
- Tool Overview: Issues an unauthorized Kerberos ticket that is valid for an arbitrary period and grants access without additional authentication.
- Tool Usage During an Attack :
  - Used to grant a host concealing a record of authentication requests access using the Golden Ticket.
  - Source host: Mimikatz execution source.
  - Destination host: The host logon in by Mimikatz.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: Not Applicable

- Service: Active Directory Domain Service

Information Acquired from log

- Source host: Execution history (Prefetch).
- Source host: Execution history (Sysmon / audit policy).
- Access history (Sysmon - RawAccessRead, audit policy - Use of important privileges).
- Destination host: Logon by an account with an illegal domain.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4673 (A privileged service was called)

    4663 (An attempt was made to access an object)

    4656 (A handle to an object was requested)

    4658 (The handle to an object was closed)

    1 (Process Create)

    5 (Process terminated)

- Destination Host

  Security Event Log

  - Event ID: 4769 (A Kerberos service ticket was requested)

    4672 (Special privileges assigned to new logon)

    4624 (An account was successfully logged on)

    4634 (An account was logged off)

Execution confirming evidence:

- Destination host: It is considered that unauthorized logon was attempted, if the following log is in the event log:
    - In the event IDs 4672, 4624, and 4634 in the event log, a logon attempt by an account with an illegal domain is recorded.

Remarks

Additional Event Logs That Can Be Output:

- At the host for which access was granted by using a Golden Ticket, logs related to the executed command may be recorded.

Test Case 12: Silver Ticket (mimikatz)

Basic Information

- Tool Name: Mimikatz (Silver Ticket)
- Category: Capturing the Domain Administrator Privilege and Account Credentials
- Tool Overview: Issues an unauthorized Kerberos ticket that is valid for an arbitrary period and grants access without additional authentication.
- Tool Usage During an Attack :
    - Used to grant a host concealing a record of authentication requests access using the Silver Ticket.

      Source host: Mimikatz execution source.

      Destination host: The host logon in by Mimikatz.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: Not Applicable
- Service: Active Directory Domain Service

Information Acquired from log

- Source host: Execution history (Prefetch).
- Source host: Execution history (Sysmon / audit policy).
- Destination host: Logon by an account with an invalid domain.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4673 (A privileged service was called)

- Destination Host

  Security Event Log

  - Event ID: 4672 (Special privileges assigned to new logon)

    4624 (An account was successfully logged on)

    4634 (An account was logged off)

Execution confirming evidence:

- Destination host: It is considered that unauthorized logon was attempted, if the following log is in the event log:
- In the event IDs 4672, 4624, and 4634 in the event log, a logon attempt by an account with an illegal domain is recorded.

Remarks

Additional Event Logs That Can Be Output:

- Security: Unlike a Golden Ticket, communication with the Domain Controller does not occur when a ticket is generated.
- At the host for which access was granted by using a Silver Ticket, logs related to the executed command may be recorded.

Test Case 13: AT Command

Basic Information

- Tool Name: AT
- Category: Command execution
- Tool Overview: Executes a task at the specified time
- Tool Usage During an Attack :
  - Used to secretly place an application or script without being recognized by the user in advance and then execute it at the desired time.

    Source host: AT command execution source.

    Destination host: The machine for which a task was registered by the AT command.

Operating Condition

- User Account Type: Administrator (Setting a task on the remote host can be performed by a standard user.)
- Targeted OS: Windows 7 / Server 2008 (The AT command was abolished in Windows 8 and later and Server 2012 and later.)
- Domain: Not required
- Communication Protocol: 445/tcp
- Service: Task Scheduler

Information Acquired from log

- Source host: Execution history (Prefetch).
- Destination host: Task creation / execution history in the task scheduler event log.
- Execution history (Sysmon/audit policy).

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    1 (Process Create)

    5 (Process terminated)

- Destination Host

  Security Event Log

  - Event ID: 4656 (A handle to an object was requested)

4663 (An attempt was made to access an object)

4658 (The handle to an object was closed)

4698 (A scheduled task was created)

4688 (A new process has been created)

106 (A task has been registered)

200 (The operation that has been started)

Execution confirming evidence:

- Source host: It is considered that a task was registered if Event ID 4689 (A process has exited) of at.exe is recorded in the event log with the execution result (return value) of "0x0".
- Destination host: It is considered that a task was executed if Event ID 106 (A task has been registered) is recorded in the event log "\Microsoft\Windows\TaskScheduler\Operational".
- The Event IDs 200 (The operation that has been started) and 201 (The operation has been completed) are registered in the event log "\Microsoft\Windows\TaskScheduler\Operational", and the return value of the Event ID 201 is set to success.

Remarks

Additional Event Logs That Can Be Output:

- Logs related to the command called from the task may be recorded.

Test Case 14: RDP

Basic Information

- Tool Name: RDP (Remote Desktop Protocol)
- Category: Remote Login
- Tool Overview: A protocol to connect to a server on which Remote Desktop Service (RDS) is running.
- Tool Usage During an Attack:
    - View files on the logged in machine.
    - Collect information (required) for connecting to other servers and clients.
    - Use as a stepping stone to connect to other equipment.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 3389/tcp
- Service: Destination host: Remote Desktop Services

Information Acquired from log

- Source host IP address Logged in user name, and success or failure of account domain connection.
- Destination host: RDP session connection start/end time and date.
- Source host: mstsc.exe execution history, file access history.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4663 (An attempt was made to access an object)

    4656 (A handle to an object was requested)

    4658 (The handle to an object was closed)

- Destination Host

  Security Event Log

  - Event ID: 4624 (An account was successfully logged on)

    21 (Remote Desktop Services: Session logon succeeded)

    24 (Remote Desktop Services: Session has been disconnected)

Execution confirming evidence:

- Destination host:
  - Event ID: 4624 is recorded in the event log.
  - Event IDs 21 and 24 are recorded in the event log "Microsoft\Windows\TerminalServices-LocalSessionManager\Operational."

Remarks

Additional Event Logs That Can Be Output:

- Depending on the environment, the following log may be recorded in the destination host event log "Security".
  - Event ID: 4624 (An account was successfully logged on).
  - Logon Type: "12".

Test Case 15: mimikatz

Basic Information

- Tool Name: Mimikatz (Remote Login)
- Category: pass-the-hash, pass-the-ticket
- Tool Overview: Executes a command with another user's privileges using a hash of the acquired password
- Tool Usage During an Attack:
  - Remotely executes a command on another machine by using a password hash for a user with Administrator privileges.

    Source host: Mimikatz execution source.

    Destination host: The destination logged in by Mimikatz.

Operating Condition

- User Account Type: Source host: Administrator

  Destination host: Privileges of the user whose hash was used
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: A random 5-digit port (WMIC)
- Service: Windows Management Instrumentation

Information Acquired from log

- Source host: Execution history (Prefetch).
- Communication logs during a remote connection.
- Process logs when a connection occurs.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    5156 (The Windows Filtering Platform has allowed a connection)

    4648 (A logon was attempted using explicit credentials)

- Destination Host

  Security Event Log

  - Event ID: 4624 (An account was successfully logged on)

    5156 (The Windows Filtering Platform has allowed a connection)

    1 (Process Create)

Execution confirming evidence:

- Destination host: It is considered that a remote login was made if:
  - The event ID 4624 is recorded in the event log "Security" regarding access from an unintended source host.

Test Case 16: MS14-058 Exploit

Basic Information

- Tool Name: MS14-058 Exploit
- Category: Escalation to SYSTEM Privileges
- Tool Overview: Executes a specified command with SYSTEM privileges
- Tool Usage During an Attack:
    - Used for a user with standard privileges to execute a command that normally requires administrator privileges.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: Not Applicable
- Service: Not Applicable

Information Acquired from log

- Execution history (Prefetch).
- The name of a process executed by the tool with SYSTEM privileges, and argument (Sysmon / audit of process tracking).

Log Generation

- Host (Windows)

  Security Event Log

    - Event ID: 4688 (A new process has been created)

4689 (A process has exited)

1 (Process Create)

5 (Process terminated)

  - Presence of Privilege Escalation at Process Execution: Process Information - Token Escalation Type.

Execution confirming evidence:

- It is considered that privilege escalation was successful if event ID: 4688 is recorded regarding a process executed with SYSTEM privileges, whose parent process cannot be the parent of the tool or that process.

Remarks

Additional Event Logs That Can Be Output:

- Other logs that are related to processes executed with SYSTEM privileges may be recorded.

Test Case 17: ntdsutil

Basic Information

- Tool Name: ntdsutil
- Category: Obtaining Active Directory database
- Tool Overview: A command to maintain Active Directory databases
- Tool Usage During an Attack :
  - Used to extract NTDS.DIT, a database for NTDS, so that the password can be analyzed using other tools (executed in Active Directory).

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows Server
- Domain: Required
- Communication Protocol: Not Applicable
- Service: Active Directory Domain Services

Information Acquired from log

- The fact that the service has started and that a driver was installed on a storage device.
- History of shadow copy creation.
- Execution history (Sysmon/audit policy).

Log Generation

- Active Directory Domain Controller

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4673 (A privileged service was called)

    8222 (Shadow copy has been created)

    4656 (A handle to an object was requested)

Execution confirming evidence:

- If ntdsutil.exe was executed the Event ID 8222 is recorded in the event log.
- A request for a handle for "[System Drive]\SNAP_[Date and Time]_VOLUME[Drive Letter]$" was successful.

- Additionally, if a log indicating that files under C:\Windows\NTDS, which cannot be normally read, were copied (Event ID: 4663) is recorded, a shadow copy was used.

Test Case 18: vssadmin

Basic Information

- Tool Name: vssadmin
- Category: Obtaining Active Directory database
- Tool Overview: Creates Volume Shadow Copy and extracts NTDS.DIT
- Tool Usage During an Attack :
    - Used to extract NTDS.DIT, a database for NTDS, so that the password can be analyzed using other tools.

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows Server
- Domain: Required
- Communication Protocol: Not Applicable
- Service: Active Directory Domain Services

Information Acquired from log

- The fact that the service has started and that a driver was installed on a storage device.
- History of shadow copy creation.
- Execution history (Sysmon/audit policy).

Log Generation

- Active Directory Domain Controller

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    8222 (Shadow copy has been created)

Execution confirming evidence:

- It is considered that a shadow copy was created if event ID 8222 is recorded in the event log.
- Additionally, if a log indicating that files under C:\Windows\NTDS, which cannot be normally read, were copied (Event ID: 4663) is recorded, a shadow copy was used.

Test Case 19: net user

Basic Information

- Tool Name: net Command (net user)
- Category: Adding or Deleting a User/Adding or Deleting a Group
- Tool Overview: Adds a user account in a client or the domain
- Tool Usage During an Attack :
  - Used to create accounts or additional sessions in the machine the attacker has infected or to communicate with other hosts.

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows

- Domain: Not required
- Communication Protocol: Not Applicable
- Service: Not Applicable

Information Acquired from log

- The fact that a user has been added is recorded in a log
- A user name and password specified by the "net user" command are recorded (Sysmon).

Log Generation

- Windows Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4656 (A handle to an object was requested)

    4720 (A user account was created)

Execution confirming evidence:

- It is considered that a user was added if event ID 4720 is recorded in the event log.

Remarks

Additional Event Logs That Can Be Output:

- If addition to a group or others were performed, the relevant access history is recorded.

Test Case 20: Find-GPOPasswords.ps1

Basic Information

- Tool Name: Find-GPOPasswords.ps1
- Category: Password and Hash Dump
- Tool Overview: Acquires any password descriptions in a group policy file.
- Tool Usage During an Attack:
    - Attempts to infiltrate other hosts using acquired passwords (by executing the tool on Active Directory).

Operating Condition

- User Account Type: Administrator (Setting a task on the remote host can be performed by a standard user.)
- Targeted OS: Windows Server
- Domain: Required
- Communication Protocol: Not Applicable
- Service: Not Applicable

Information Acquired from log

- Execution history (Prefetch).
- The information is not of use when PowerShell is used in regular operations.
- The fact that PowerShell was started is recorded.
- The fact that a file in which passwords are dumped (GPPDataReport-[Domain Name]-[Time and Date].csv) is output is recorded.

Log Generation

- Active Directory Domain Controller (Windows Server)

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4656 (A handle to an object was requested)

    4663 (An attempt was made to access an object)

    4658 (The handle to an object was closed)

    4660 (An attempt was deleted)

    1 (Process Create)

    5 (Process terminated)

- Object: -Object Name: ("GPPDataReport-[Domain Name]-[Time and Date].csv")

Execution confirming evidence:

- A file in which a password was dumped (GPPDataReport-[Domain Name]-[Time and Date].csv) is output.

Test Case 21: csvde

Basic Information

- Tool Name: csvde
- Category: Acquisition of Account Information
- Tool Overview: Outputs account information on the Active Directory in the CSV format
- Tool Usage During an Attack :

    - Used to extract information on an existing account and select users and clients available as attack targets.

      Source host: csvde command execution source.

      Destination host: The machine in which information is collected by the csvde command.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Required
- Communication Protocol: 389/tcp
- Service: Active Directory Domain Services

Information Acquired from log

- Source host: Execution history (Prefetch).
- Source host: The fact that a csv file was created by csvde.exe.
- Destination host: Inbound to 389/tcp and login with Kerberos Authentication are recorded.

Log Generation

- Source Host

  Security Event Log

    - Event ID: 4688 (A new process has been created)

      4689 (A process has exited)

      5156 (The Windows Filtering Platform has allowed a connection)

      4663 (An attempt was made to access an object)

4656 (A handle to an object was requested)

4658 (The handle to an object was closed)

1 (Process Create)

5 (Process terminated)

- Destination Host

  Security Event Log

  - Event ID: 5156 (The Windows Filtering Platform has allowed a connection)

    4624 (An account was successfully logged on)

    4634 (An account was logged off)

Execution confirming evidence:

- Source host: csvde.exe was executed and a file specified by the "-f" option was created.
- C:\Users\[User Name]\AppData\Local\Temp\csv[Random Number].tmp was created and deleted.

Test Case 22: ldifde

Basic Information

- Tool Name: ldifde
- Category: Acquisition of Account Information
- Tool Overview: Outputs account information on the Active Directory in the LDIF format
- Tool Usage During an Attack :

  - Used to extract information on an existing account and select users and clients available as attack targets.

    Source host: ldifde command execution source.

    Destination host: The machine in which information is collected by the ldifde command.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows
- Domain: Required
- Communication Protocol: 389/tcp
- Service: Active Directory Domain Services

Information Acquired from log

- Source host: Execution history (Prefetch).
- Source host: The fact that a LDIF file was created by ldifde.exe.
- Destination host: Inbound to 389/tcp and login with Kerberos Authentication are recorded.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    5156 (The Windows Filtering Platform has allowed a connection)

    4656 (A handle to an object was requested)

4663 (An attempt was made to access an object)

4658 (The handle to an object was closed)

1 (Process Create)

5 (Process terminated)

- Destination Host

  Security Event Log

  - Event ID: 5156 (The Windows Filtering Platform has allowed a connection)

    4624 (An account was successfully logged on)

Execution confirming evidence:

- Source host: ldifde.exe was executed and a file specified by the "-f" option was created.

Test Case 23: timestomp

Basic Information

- Tool Name: timestomp
- Category: Deleting Evidence
- Tool Overview: Changes the file timestamp
- Tool Usage During an Attack :
  - Used to conceal the access to the file by restoring the timestamp which has changed as a result of the use of file by the attacker.

Operating Condition

- User Account Type: Standard user
- Targeted OS: Windows

- Domain: Not required
- Communication Protocol: Not Applicable
- Service: Not Applicable

Information Acquired from log

- Execution history (Prefetch).
- Execution history (Sysmon/audit policy).
- Auditing of change of file creation date and time.

Log Generation

- Host Windows

  Security Event Log

  - Event ID: 4688 (A new process has been created)

    4689 (A process has exited)

    4656 (A handle to an object was requested)

    4663 (An attempt was made to access an object)

    4658 (The handle to an object was closed)

    1 (Process Create)

    2 (File creation time changed)

    9 (RawAccessRead detected - Direct disk read detected)
- Event ID: 2 shows a change in the file creation date and time, but it is output regardless of the type (creation, change, or access) of the changed timestamp. If an item other than the creation date and time is changed, the same time (original date and time) is recorded in the timestamp before and after the change.

Execution confirming evidence:

- It is considered that the timestamp was changed if event ID: 4663 is recorded in the event log "Security", and the "WriteAttributes" keyword for the target file is set to "Audit Success".

Test Case 24: wevtutil

Basic Information

- Tool Name: wevtutil
- Category: Deleting Event Log
- Tool Overview: Deletes Windows event logs
- Tool Usage During an Attack :
  - Used to delete the evidence of an attack.

    Source host: wevtutil command execution source.

    Destination host: The machine accessed by the wevtutil command.

Operating Condition

- User Account Type: Administrator
- Targeted OS: Windows
- Domain: Not required
- Communication Protocol: 135/tcp
- Service: Event Log

Information Acquired from log

- The fact that an event log was cleared remains in each cleared log of the host.

- The account used for clearing logs and the host that executed the clear command can be confirmed.

Log Generation

- Source Host

  Security Event Log

  - Event ID: 4688 (A new process has been created)
    4689 (A process has exited)
    4648 (A logon was attempted using explicit credentials)
    1 (Process Create)
    5 (Process terminated)

- Destination Host

  Security Event Log

  - Event ID: 104 (The System log file was cleared)

Execution confirming evidence:

- Source host: It is considered that logs were cleared if event ID: 104 is recorded in each target event log.

Results

The summary of analysis of all the above-mentioned test cases, are explained in two tables below. These tables will help in defining guidelines for creating filters in ELK stack which will help in detection of lateral movement or the analysis of the test case can act like an example to analyze incidents. The tables below explain

- How tools are used to establish lateral movement in a network.
- Proof of Execution of tools.

TABLE 1: How tools are used to establish lateral movement in a network

TOOL	Technique	Tool usage in an attack
PsExec	PsExec	Used to remotely execute a command on client and servers in a domain. Source host: PsExec command execution source Destination host: The destination logged in by the PsExec command
PowerShell	PowerShell	Used to change settings to enable the Domain Controller and other hosts on the network to perform operations requiring administrator rights. Source host: PowerShell command execution source Destination host: The destination logged in by the PowerShell command
WinRM	Windows Remote Management	Used for carrying out an investigation before executing a remote command. Source host: WinRM command execution source Destination host: The machine accessed by the WinRM command
WinRS	Windows Remote Management	This tool is sent by the BITS and remotely executed using winrs. Source host: WinRS command execution source Destination host: The machine accessed by the WinRS command

wmic	WMI	Used to acquire information on the remote system or to execute a command with WMI. Source host: wmic command execution source Destination host: The host accessed by the wmic command
wmiexec.vbs	WMI	Executes a script for other hosts. Source host: The source that executes wmiexec.vbs Destination host: The machine accessed by the wmiexec.vbs
PWDumpX	Pass-the-hash	Uses the acquired hash to perform attacks such as pass-the-hash. Source host: PWDumpX execution source Destination host: The destination logged in by PWDumpX
Quarks PwDump	Pass-the-hash	Uses acquired hash information to perform logon authentication on other hosts
WCE	Pass-the-hash	Uses the acquired hash information to perform pass-the-hash and other attacks.
WCE	Pass-the-hash, Pass-the-ticket	Remotely executes a command on another machine by using a password hash for a user with Administrator privileges which belongs to Active Directory Source host: WCE execution source Destination host: The destination logged in by WCE

Golden Ticket (mimikatz)	Pass the Ticket	Used to grant a host concealing a record of authentication requests access using the Golden Ticket. Source host: Mimikatz execution source Destination host: The host logon in by Mimikatz
Silver Ticket (mimikatz)	Pass the Ticket	Used to grant a host concealing a record of authentication requests access using the Silver Ticket. Source host: Mimikatz execution source Destination host: The host logon in by Mimikatz
AT	Scheduled tasks	Used to secretly place an application or script without being recognized by the user in advance and then execute it at the desired time. Source host: AT command execution source Destination host: The machine for which a task was registered by the AT command
RDP	Remote Login/Remote desktop protocol	View files on the logged in machine Collect information (required) for connecting to other servers and clients Use as a stepping stone to connect to other equipment

mimikatz	WMI/Remote Login	Remotely executes a command on another machine by using a password hash for a user with Administrator privileges Source host: Mimikatz execution source Destination host: The destination logged in by Mimikatz
MS14-058 Exploit	Escalation to SYSTEM Privileges	Used for a user with standard privileges to execute a command that normally requires administrator privileges
ntdsutil	Active Directory	Used to extract NTDS.DIT, a database for NTDS, so that the password can be analyzed using other tools (executed in Active Directory).
vssadmin	Active Directory	Used to extract NTDS.DIT, a database for NTDS, so that the password can be analyzed using other tools.
net user	Active Directory	Used to create accounts or additional sessions in the machine the attacker has infected or to communicate with other hosts.
Find-GPOPasswords.ps1	Active Directory	Attempts to infiltrate other hosts using acquired passwords (by executing the tool on Active Directory).

csvde	Active Directory	Used to extract information on an existing account and select users and clients available as attack targets. Source host: csvde command execution source Destination host: The machine in which information is collected by the csvde command
ldifde	Active Directory	Used to extract information on an existing account and select users and clients available as attack targets. Source host: ldifde command execution source Destination host: The machine in which information is collected by the ldifde command
timestomp	Deleting Evidence	Used to conceal the access to the file by restoring the timestamp which has changed as a result of the use of file by the attacker
wevtutil	Deleting Event Log	Used to delete the evidence of an attack. Source host: wevtutil command execution source Destination host: The machine accessed by the wevtutil command

TABLE 2: Proof of Execution of tools

TOOL	Proof of Execution
PsExec	Confirmation of PsExec was execution: Source host: Event ID 4689 (A process has exited) of psexec.exe was recorded in the event log with the execution result (return value) of "0x0". Destination host: PSEXESVC.exe is installed.
PowerShell	If the following logs that have the same log time are found, it confirms that a remote command was executed. This also applies to Prefetch. Source host: Event ID 4689 (A process has exited) of PowerShell was recorded in the event log with the execution result (return value) of "0x0". Destination host: Event ID 4689 (A process has exited) of wsmprovhost.exe was recorded in the event log with the execution result (return value) of "0x0".
WinRM	Source host: A log indicating that cscript.exe accessed the destination host with Event IDs 1 and 5 of the event log.

WinRS	The execution of WinRS is recorded in the event log "Application and Service\Microsoft\Windows\Windows Remote Management\Operational".
wmic	If the following logs that have the same log time are found at source host and destination host, a remote connection was made: Source host: Event ID 4689 (A process has exited) of WMIC.exe was recorded in the event log with the execution result (return value) of "0x0". Destination host: It is recorded in the event log that WmiPrvSE.exe was executed with the event IDs 1 and 5.
wmiexec.vbs	Destination host: The "WMI_SHARE" share has been created and deleted.
PWDumpX	Source host: If "[Path to Tool]\[Destination Address]-PWHashes.txt" has been created, it was successfully executed.
Quarks PwDump	A temporary file ("SAM-[Random Number].dmp") was created and deleted.
WCE	The "C:\Users\[User Name]\AppData\Local\Temp\wceaux.dll" file was created and deleted.

WCE	Source host: The fact that WCESERVICE was installed and executed is recorded. Destination host: The fact that a logon was made from a remote host is recorded. Both source host and destination host: The fact that communication using WMI occurred is recorded.
Golden Ticket (mimikatz)	Destination host: It is considered that unauthorized logon was attempted, if the following log is in the event log: In the event IDs 4672, 4624, and 4634 in the event log, a logon attempt by an account with an illegal domain is recorded.
Silver Ticket (mimikatz)	Destination host: It is considered that unauthorized logon was attempted, if the following log is in the event log: In the event IDs 4672, 4624, and 4634 in the event log, a logon attempt by an account with an illegal domain is recorded.

AT	Source host: It is considered that a task was registered if Event ID 4689 (A process has exited) of at.exe is recorded in the event log with the execution result (return value) of "0x0". Destination host: It is considered that a task was executed if Event ID 106 (A task has been registered) is recorded in the event log "\Microsoft\Windows\TaskScheduler\Operational". The Event IDs 200 (The operation that has been started) and 201 (The operation has been completed) are registered in the event log "\Microsoft\Windows\TaskScheduler\Operational", and the return value of the Event ID 201 is set to success.
RDP	Destination host: Event ID: 4624 is recorded in the event log. Event IDs 21 and 24 are recorded in the event log "Microsoft\Windows\TerminalServices-LocalSessionManager\Operational"
mimikatz	Destination host: It is considered that a remote login was made if: The event ID 4624 is recorded in the event log "Security" regarding access from an unintended source host.
MS14-058 Exploit	It is considered that privilege escalation was successful if event ID: 4688 is recorded regarding a process executed with SYSTEM privileges, whose parent process cannot be the parent of the tool or that process.

ntdsutil	If ntdsutil.exe was executed the Event ID 8222 is recorded in the event log. A request for a handle for "[System Drive]\SNAP_[Date and Time]_VOLUME[Drive Letter]$" was successful Additionally, if a log indicating that files under C:\Windows\NTDS, which cannot be normally read, were copied (Event ID: 4663) is recorded, a shadow copy was used.
vssadmin	It is considered that a shadow copy was created if event ID 8222 is recorded in the event log. Additionally, if a log indicating that files under C:\Windows\NTDS, which cannot be normally read, were copied (Event ID: 4663) is recorded, a shadow copy was used.
net user	It is considered that a user was added if event ID 4720 is recorded in the event log.
Find-GPOPasswords.ps1	A file in which a password was dumped (GPPDataReport-[Domain Name]-[Time and Date].csv) is output.
csvde	Source host: csvde.exe was executed and a file specified by the "-f" option was created. C:\Users\[User Name]\AppData\Local\Temp\csv[Random Number].tmp was created and deleted.

ldifde	Source host: ldifde.exe was executed and a file specified by the "-f" option was created.
timestomp	It is considered that the timestamp was changed if event ID: 4663 is recorded in the event log "Security", and the "WriteAttributes" keyword for the target file is set to "Audit Success".
wevtutil	Source host: It is considered that logs were cleared if event ID: 104 is recorded in each target event log.

Conclusion and Future Opportunities

Under current circumstances where it is difficult to prevent initial infiltration of a network, it is important to always consider and improve the method for acquiring logs to detect an ongoing incident. This further helps in analyzing the amount of damage after an incident occurs to prevent the spread of damage and review post-incident security measures. Early detection and accurate response to targeted attacks, are surely important to reduce damage.

Many tools do not leave evidence of having been executed with the default Windows settings or many attackers cover their tracks, which may cause a delayed incident response or may leave incident investigations unresolved. To analyze what the attacker did in detail, an environment that allows for more logs to be collected than those obtained with the default settings needs to be prepared in advance as suggested in this document.

Once a proper logging system like ELK stack is set up, creation of filters based on the understanding developed from the findings mentioned in the test cases can help in detection of lateral movement. This research takes into consideration, majority of tools and techniques used by adversaries to move in a windows-based network laterally. It showcases the execution of these tools and suggests how usage of such tools can be detected using logging. Once the adversary starts attempting to do lateral movement using tools or techniques mentioned in this research, simple filters in ELK stack can help detect those attempts by recognizing the pattern of events generated by the tool. Once detected, an organization can document it and run incident response to minimize the damage. Hence, this research gives a solution to detect lateral movements, by using ELK stack as a logging tool.

Future opportunities related to this work will be:

- Including Linux and MAC based environments.
- Incorporating the above-mentioned detection technique to an IDS (Intrusion detection system) to create real time alerts or even an IPS (Intrusion prevention system).
- Incorporating solutions related to other tools used for lateral movement which are not covered in this research.

References

Abdelkader Lahmadi, F. B. (2015, 9 May). *Powering Monitoring Analytics with ELK stack.* Retrieved from https://hal.inria.fr/hal-01212015/document

Alert Logic Staff. (2016, December 30). *The Cyber Kill Chain: Understanding Advanced Persistent Threats*. Retrieved from www.alertlogic.com: https://www.alertlogic.com/blog/the-cyber-kill-chain-understanding-advanced-persistent-threats/

Bandos, T. (2017, July 27). *SEEK EVIL, AND YE SHALL FIND: A GUIDE TO CYBER THREAT HUNTING OPERATIONS.* Retrieved from digitalguardian.com: https://digitalguardian.com/blog/seek-evil-and-ye-shall-find-guide-cyber-threat-hunting-operations

Banks, J. T. (2017, July 10). *How To Do Endpoint Monitoring on a Shoestring Budget – Webcast Write-Up.* Retrieved from www.blackhillsinfosec.com: https://www.blackhillsinfosec.com/endpoint-monitoring-shoestring-budget-webcast-write/

Brower, J. (2015, March 19). *Using Sysmon to Enrich Security Onion's Host-Level Capabilities*. Retrieved from digital-forensics.sans.org: https://digital-forensics.sans.org/community/papers/gcfa/sysmon-enrich-security-onions-host-level-capabilities_10612

Devry, J. (n.d.). *How Attackers Lay the Groundwork for Lateral Movement*. Retrieved from cybersecurity-insiders.com: https://www.cybersecurity-insiders.com/how-attackers-lay-the-groundwork-for-lateral-movement/

Devry, J. (n.d.). *THREAT HUNTING FOR LATERAL MOVEMENT*. Retrieved from cybersecurity-insiders.com: https://www.cybersecurity-insiders.com/threat-hunting-for-lateral-movement/

Dunwoody, M. (2016, February 11). *Greater Visibility Through PowerShell Logging*. Retrieved from www.fireeye.com: https://www.fireeye.com/blog/threat-research/2016/02/greater_visibilityt.html

Eric D. Knapp, J. T. (n.d.). *Industrial Network Security*. ELSEVIER.

HARARI, E. (2017, July 3). *Lateral Movement and Threat Actors – Watch your network!* Retrieved from verint.com: https://cyber.verint.com/lateral-movement-and-threat-actors/

Heller, M. (2017, September 28). *Network lateral movement from an attacker's perspective*. Retrieved from searchsecurity.techtarget.com: http://searchsecurity.techtarget.com/news/450427135/Network-lateral-movement-from-an-attackers-perspective

Hermanowski, D. (n.d.). *Open Source Security Information Management System Supporting IT Security Audit.* Retrieved from www.wil.waw.pl: https://www.wil.waw.pl/art_prac/2015/pub_cybconfCybersec15_DH-OSSIM-ieee_REVIEW_RC05_ver_PID3720933.pdf

Hibbert, B. (2017, May 2). *10 Steps to Stop Lateral Movement in Data Breaches*. Retrieved from www.beyondtrust.com: https://www.beyondtrust.com/blog/10-steps-stop-lateral-movement-data-breaches/

Hosburg, M. (2017, July 6). *offensive intrusion analysis uncovering insiders threat hunting active defense*. Retrieved from www.sans.org: https://www.sans.org/readingroom/whitepapers/detection/offensive-intrusion-analysis-uncovering-insiders-threat-hunting-active-defense-37885

JPCERT Coordination Center. (2017, June 12). *Detecting Lateral Movement throuh Tracking Event Logs.* Retrieved from www.jpcert.or.jp: https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf

JPCERT/CC. (2017, December 5). *Detecting Lateral Movement through Tracking Event Logs (Version 2).* Retrieved from www.jpcert.or.jp: https://www.jpcert.or.jp/english/pub/sr/Detecting%20Lateral%20Movement%20through%20Tracking%20Event%20Logs_version2.pdf

Keith Stouffer, V. P. (2015). *Guide to Industrial Control Systems (ICS) Security.* National Institute of Standards and Technology.

Kvetch. (2017, February 4). *Analysis of DCOM Lateral Movement Using MMC20.Application*. Retrieved from thenegative.zone: http://thenegative.zone/incident%20response/2017/02/04/MMC20.Application-Lateral-Movement-Analysis.html

LEWIS, J. (2015, May 25). *DETECTING ADVANCED THREATS WITH SYSMON, WEF AND ELASTICSEARCH.* Retrieved from ROOT9B: https://www.root9b.com/sites/default/files/whitepapers/R9B_blog_005_whitepaper_01.pdf

M.SORIA-MACHADO, D. C. (2017, February 27). *Detecting Lateral Movements in Windows Infrastructure.* Retrieved from cert.europa.eu: http://cert.europa.eu/static/WhitePapers/CERT-EU_SWP_17-002_Lateral_Movements.pdf

MITRE. (2017, August 14). *LATERAL MOVEMENT*. Retrieved from https://attack.mitre.org/: https://attack.mitre.org/wiki/Lateral_Movement

MITRE. (2018, January 10). *Access Token Manipulation*. Retrieved from attack.mitre.org: https://attack.mitre.org/wiki/Technique/T1134

MITRE. (2018, January 11). *Logon Scripts*. Retrieved from attack.mitre.org: https://attack.mitre.org/wiki/Technique/T1037

MITRE. (2018, February 21). *Replication Through Removable Media*. Retrieved from attack.mitre.org: https://attack.mitre.org/wiki/Technique/T1091

M-Labs. (2012, November 7). *An In-Depth Look Into Data Stacking*. Retrieved from www.fireeye.com: https://www.fireeye.com/blog/threat-research/2012/11/indepth-data-stacking.html

National Security Agency/Central Security Service . (2013, February 28). *Spotting the Adversary with Windows Event Log Monitoring.* Retrieved from cryptome.org: https://cryptome.org/2014/01/nsa-windows-event.pdf

Nolette, R. (2017, August 16). *FINDING EVIL WHEN HUNTING FOR LATERAL MOVEMENT*. Retrieved from sqrrl.com: https://sqrrl.com/finding-evil-when-hunting-for-lateral-movement/

Ram Shankar Siva Kumar, N. S. (2017). *United States of America Patent No. US9591006.* Retrieved from https://www.google.com/patents/US9591006

Rodriguez, R. (2017, March 11). *Chronicles of a Threat Hunter: Hunting for In-Memory Mimikatz with Sysmon and ELK - Part I (Event ID 7)*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/03/chronicles-of-threat-hunter-hunting-for.html

Rodriguez, R. (2017, March 22). *Chronicles of a Threat Hunter: Hunting for In-Memory Mimikatz with Sysmon and ELK - Part II (Event ID 10)*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/03/chronicles-of-threat-hunter-hunting-for_22.html

Rodriguez, R. (2017, April 1). *Chronicles of a Threat Hunter: Hunting for In-Memory Mimikatz with Sysmon, Win Event Logs, and ELK - Part III (Overpass-the-Hash - EIDs 10, 4624, 4648, 4768)*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/04/chronicles-of-threat-hunter-hunting-for.html

Rodriguez, R. (2017, April 11). *Chronicles of a Threat Hunter: Hunting for Remotely Executed Code via Services & Lateral Movement with Sysmon, Win Event Logs, and ELK*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/04/chronicles-of-threat-hunter-hunting-for_11.html

Rodriguez, R. (2017, March 26). *Chronicles of a Threat Hunter: Hunting for WMImplant with Sysmon and ELK - Part I (EID 1,12, 13, 17 & 18)*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/03/chronicles-of-threat-hunter-hunting-for_26.html

Rodriguez, R. (2017, March 3). *Cyber Wardog Lab*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/03/building-sysmon-dashboard-with-elk-stack.html

Rodriguez, R. (2017, June 28). *Enabling Enhanced PowerShell logging & Shipping Logs to an ELK Stack for Threat Hunting*. Retrieved from cyberwardog.blogspot.com: https://cyberwardog.blogspot.com/2017/06/enabling-enhanced-ps-logging-shipping.html

Ryan Nolette, A. F. (2017, June 7). *How to Hunt for Lateral Movement on Your Network*. Retrieved from www.slideshare.net: https://www.slideshare.net/sqrrl/how-to-hunt-for-lateral-movement-on-your-network

Scutt, M. (2015, May 27). *Detecting Lateral Movement with Windows Event Logs*. Retrieved from www.rapid7.com: https://www.rapid7.com/resources/using-windows-event-logs-to-detect-lateral-movement/

Security Zap. (n.d.). *Vulnerabilities in Industrial Control Systems – SCADA*. Retrieved from https://securityzap.com: https://securityzap.com/vulnerabilities-in-industrial-control-systems/

SMOKESCREEN. (2016, August). *The top 20 Lateral Movement tactics*. Retrieved from www.smokescreen.io: https://www.smokescreen.io/wp-content/uploads/2016/08/Top-20-Lateral-Movement-Tactics.pdf

step6: loading sample kibana dashboards. (n.d.). Retrieved from https://www.elastic.co/: https://www.elastic.co/guide/en/beats/winlogbeat/1.2/_step_6_loading_sample_kibana_dashboards.html

Wells, D. (2015, September 17). *Spotting the Adversary with Windows Event Log Monitoring - I*. Retrieved from www.redblue.team: http://www.redblue.team/2015/09/spotting-adversary-with-windows-event.html

Wells, D. (2015, September 21). *Spotting the Adversary with Windows Event Log Monitoring, Part II*. Retrieved from www.redblue.team: http://www.redblue.team/2015/09/spotting-adversary-with-windows-event_21.html

Williamson, W. (2016, April 11). *Lateral Movement: When Cyber Attacks Go Sideways*. Retrieved from www.securityweek.com: http://www.securityweek.com/lateral-movement-when-cyber-attacks-go-sideways

WMI, L. M. (2017, November 20). *Tony Lambert*. Retrieved from www.redcanary.com: https://www.redcanary.com/blog/lateral-movement-winrm-wmi/

Zarzosa, S. G. (2017, February 28). *DiSIEM – Diversity-enhancements for SIEMs.* Retrieved from disiem-project.eu: http://disiem-project.eu/wp-content/uploads/2017/03/D2.1.pdf