

# Automated Threat Hunting Using ELK Stack – A Case Study

MOZA AL SHIBANI

Ibri College of Technology, Ministry of Manpower, Ibri, Sultanate of Oman  
Moza.Alshebani@ibriict.edu.om

E ANUPRIYA

Ibri College of Technology, Ministry of Manpower, Ibri, Sultanate of Oman  
Dr.e.anupriya@ibriict.edu.om

**Abstract - Modern threats are very much sophisticated and they bypass legitimate security tools. Static threat hunting methods are futile. The alternate threat hunting method is to dynamically analyze their entry and behavior in the network. The two popular methods to analyze threats are to use smart machine intelligent hunting software or monitor end point activity. The end point activities can be obtained from system log using Sysmon. The event logs are filtered to eliminate the normal day-to-day activities and the suspicious activities are forwarded to server with ELK stack. The server analyzes the process creation, parent processes and their behavior. Filter is applied on the server side to analyze and hunt the threats. As a case study, threats like 1. Malicious code to remotely access files on shared drive and to delete them 2. Remote registry access to create or delete files on victim's registry 3. Malware codes to escalate rights and to delete files were injected on the victim client machine by a threat actor from another client. The system identified all the threats successfully and segmented them with alert message. The complete system was implemented on virtual environment on Windows with Oracle VM Virtual Box for creating virtual environment.**

## 1. Introduction

In the modern world most of our day to day work, activities and procedures are automatized. The use of organization's network or use of internet has become an inevitable resource of work. On the other hand, technology developments have aided threat actors to use newer, stealthier ways to invade network and to gain persistence in the network. They use obfuscation techniques to defame or damage or to obtain ransom benefit. The general threats include computer virus, security software, Trojan horse, Adware and Spyware, Computer Worm, Denial of Service (DoS), Distributed Denial of Service (DDoS), Phishing, Rootkit, SQL Injection, and Man in the Middle Attacks (MITM). In the current statistical surface, every system is infected by one or other type of Malware.

Malware is a generic term which covers all kinds of threats. Malicious software or in other term Malware in current digital world are modernized. Contrary to traditional malware, modern malware are highly stealthy focusing primarily on unknown vulnerabilities of the network. These modern malware attack randomly or on specific targets. The modern malware are well defined with clear goal. Due to these characteristics, the network defenses of today like Intrusion prevention system (IPS), antivirus software, firewalls, cloud led alternatives and virtual private networks are no longer sufficient to prevent modern malware entry and persistence in the network.

Firewalls can inspect and monitor ports but cannot inspect communications that are happening through the ports. IDS/IPS prevents attacks with the help of signatures of known threats. Though literature states that IDS/IPS prevents unknown threats, in reality it is not the case. Unknown threats can be addressed only with rich understanding of vulnerability. Hence, known and unknown vulnerabilities play a vital role in the plane of threats. This necessitates anti-malware to be smart, intelligent enough to critically analyze not only the network traffic but also the processes that spawn them. Signature or list based security alternatives alone may not be effective.

Also, in the current scenario of threats, threat actors use newer, stealthier to invade network and to gain persistence in the network. A plausible and easy way for threat actors is to repudiate as legitimate software like PowerShell and Windows Management Instrumentation. By using these tools, the modern malware deceit antivirus software and include themselves in whitelist of a network. This kind of threats are stateless and fileless. Once such threats are fileless malware variant of modern malware.

## 2. Literature Review

The recent developments in the information security sector are to deploy smart and intelligent solutions to threats. Pacing with the development, the adversary counter parts, the threat actors are adopting even more smart and intelligent methods, tools and software to build sophisticated threats which bypasses the legitimate security tools and software to gain entry and persistence in the network.

This literature survey discusses the different work carried out by researchers in the field of threat hunting.

[Hao et al 2014] points out the conventional malware detection techniques are not efficient enough to confront the booming malware variants. They propose a fairly novel approach to dynamically identify malware and the outcome triggering. The authors use multipath execution techniques to explore malware and their resulting outcomes which they call as Malicious Behaviors and Outcomes (MBO). Further the authors have used virtual monitor to extract the features of malicious behavior to model a classifier. The classifier can predict new malware variants with low false positives and identify malware with obfuscation techniques.

[Ismahan et al 2014] have presented content based classification approach instead of conventional signature based approach. The authors talk about stateless (at packet level) and stateful level (using TCP byte stream) malware. The success of any classifier depends on the observed training samples. The authors have incorporated Snort malware signature in Naïve Bayes Model to train the classifier which enables low featured search space and effective detection of malware at the packet level (Stateless). The authors claim to detect malware in transit itself before it reaches the endpoint. Further, their work states to viability to detect both stateless and stateful malware.

[Brown et al 2017] discusses the limitation of local network defenders for preventing cyber intrusions and their inability to equip with adequate tools to combat cyber intrusions. The authors suggest the network-defense data sources like log files, endpoint artifacts and network traffic could be the key sources to identify threats. They propose a next generation platform know as Comprehensive Hunt and Ultimate Cyber Kit (CHUCK) which enables detection of new threats in timely manner using network-defense data sources.

[Akshatha et al 2017] states the current situation of existence of number of variants of Malware makes static detection methods futile. The authors proposes a dynamic malware analysis approach to automatically detect malware by their change in behavior in a controlled environment. The authors use data mining techniques like classification and clustering techniques to report the malware identification in the system. Also, the authors have used F-measure to support their quality of detection.

[Leandro et al 2018] have presented a very comprehensive report on Fileless Threats, Analysis of Fileless Threats and their detection. Leandro et al have identified two ways to detect Fileless threats 1: Using Anti-Malware Scan Interface (AMSI), which can scan memory, files and malicious content to detect fileless malware. In this case, the PowerShell is implemented with AMSI. However, the modern malware are capable of bypassing AMSI and therefore the author have pointed out this technique to be invincible. 2. Monitoring End point activities, which collects log from different systems to analyze and detect fileless threats. The normal log cannot provide enough information about system activities. Therefore, the authors have proposed the use of Sysmon or Endpoint Detection and Response (EDR). The authors have employed different open source projects and free tools to implement monitoring.

[Vasileios et al 2018] present their work exhibiting the existing trade between the adversaries of attackers and intelligence programs in response to it. They have proposed automated system to hunt threats using Sysmon log and classify the threats in different levels based on the identified characteristics. The different threat classification levels are High, Medium, Low and Unknown. The authors have employed continuous updated threat intelligence using Ontology and have developed course of actions in response to compromises.

Literature review comprehensively presents the inability of the static security tools and emphasizes the requirement of intelligent machine learning tools to automatically identify threats. The local network defenders should equip with sophisticated tools to leverage stateless threats. Modern threats use files and obfuscation techniques in a stealthier way to destroy the victim.

Section 3 provides an overview of the proposed system. Section 4 discusses the system implementation along with the configuration settings.

## 3. Proposed system

The alternates to combat threats are deploying intelligent anti-malware or monitoring endpoint's activities. The network's endpoint activities monitoring provides us with rich information on processes, parent processes and process behavior with which the abnormal activities can be filtered out to find malicious processes. This paper focuses on detecting the threats by monitoring the network endpoint's activities. A virtual environment is set with Client-Server configuration to realize the Automated Threat Hunting System (ATHS). Oracle VM Virtual Box is used in this work to simulate the virtual environment.

The previous discussion clearly states the behavior of malicious software and the process that spawned them should be monitored to leverage the threat alert. System logs are the potential source for endpoint activities. The activities at endpoints can be accumulated from different system logs. However, the normal logs do not provide sufficient information about system processes, parent processes and their behavior. The alternate solutions for getting system processes and their behavior can be obtained through SYSMON log or EDR applications. In this project, Sysmon is used to collect all generated system logs.

Sysmon (System Monitoring) is a Windows based service, which can record every incident in the network. Filtering rule is applied to Sysmon log to filter out normal processes and feed the suspicious activities to Server with ELK stack to hunt for malicious activities and to raise alert. Hunting for exactly malicious process or activity is really challenging task as the results may contain many false positives. Elastic search default rules are applied to hunt malicious behavior.

Threats like 1. Malicious code to remotely access files on shared drive and deleting those 2. Remote registry access to create or delete files on victim's registry 3. Malware codes to escalate rights and to delete files are proposed to be injected on the victim client machine by a threat actor from another client and evaluate if threats are identified, captured and alerted in kibana display.

The primary objectives of this paper are

- 1) To extract only interested log events from Sysmon log
- 2) To differentiate normal and abnormal events from Sysmon log
- 3) To segment events into classes using clustering process
- 4) To identify the outliers and report malicious events
- 5) To segment them, learn automatically and predict further events.

The features of the proposed system includes 1: Hunt for new sophisticated threats automatically using their dynamic behavior, 2: Segment them and initiate course of action and raise alert to administrators, 3: Visualize threat behavior dynamically.

#### 4. System Implementation

Threat actors or Malware attack a system through End Points of the Network. The system developed in this project is a host based threat detection system which is implemented through virtual environment. The host based threat detection system architecture is shown in figure 1 and virtual implementation in figure 2. The system is emulated as a Client-Server Model. The basic system requirements to realize the architecture is

1. Single Node or Full ELK stack
2. Appropriate Input / Output filter in Logstash
3. Windows VM or Windows Endpoint

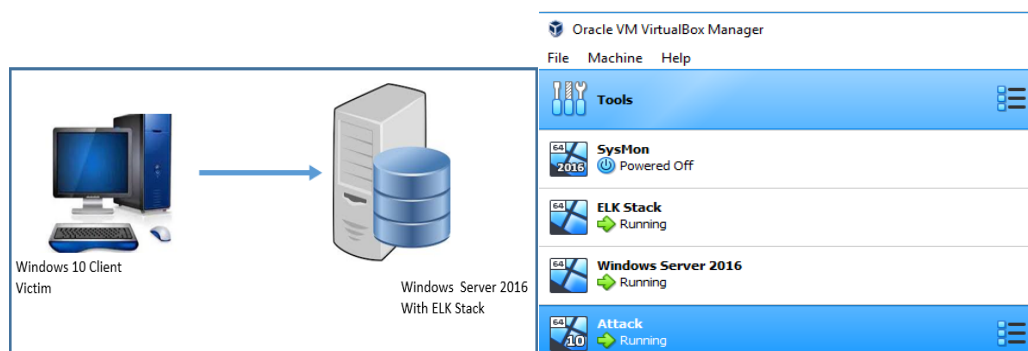


Fig.1. Single client connected to server in ATHS Fig. 2. Virtual Implementation of ATHS

Oracle Virtual Box is a virtualization product which can be used to build a virtual lab with nodes and artifacts. Oracle VM Virtual Box is rich in features and has high performance. It is an Open Source Software under GNU General Public License (GPL) version 2. It can run on Windows, Macintosh, Solaris and Linux hosts. In this project, it is implemented on Windows. One virtual client and one virtual server node is simulated in the virtual lab.

1. Nodes are the servers and we have used one Server node with ELK stack.
2. Artifacts (Software Components)

Different artifacts are installed on Client and Server as shown in Table 1.

Table 1. Artifacts – Client and Server

Client	Server
Sysmon	ELK Stack: Elastic Search, Logstash, Kibana
Winlogbeat	Java

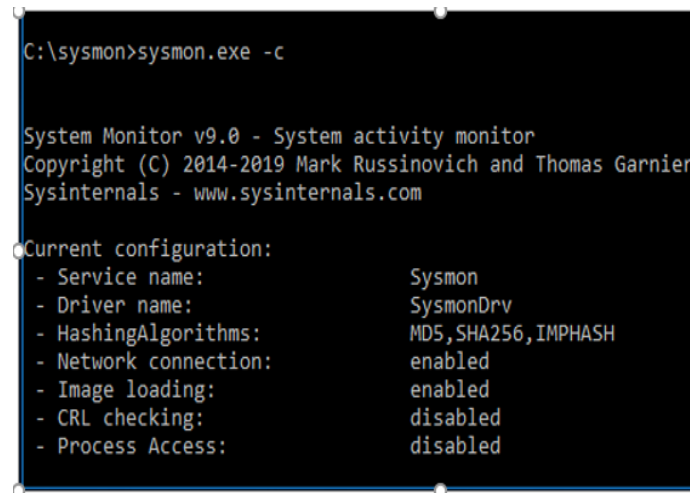
#### 4.1. Client Side Implementation

Client side implementation is carried out in three steps.

1. Install Sysmon on client VM and configure it
2. Apply advanced filtering to extract only suspicious event logs.
3. Use Data shipper, Winlogbeat to transport suspicious event log to Server

##### Step 1: Sysmon Installation and Configuration on VM Client.

Sysmon can be installed and configured with different options. In this project, the Sysmon is installed with option to include # hash algorithms. The configuration settings is in shown in figure 3.



```
C:\sysmon>sysmon.exe -c

System Monitor v9.0 - System activity monitor
Copyright (C) 2014-2019 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: MD5,SHA256,IMPHASH
- Network connection: enabled
- Image loading: enabled
- CRL checking: disabled
- Process Access: disabled
```

Fig. 3. Sysmon Basic Configuration

##### Step 2: Advanced Filtering of Sysmon Log using XML

The Sysmon configuration with basic option do not support disabling events and hence advance configuration is required to enable or disable events. Therefore, Sysmon is configured with advance filter settings using XML in config.xml file.

Event logs are highly noisy since it contains high volumes of events recorded. Perhaps, all events are not of our interest. The important task is to monitor the behavior of malicious processes that gain entry in to the network and trying to sustain in the network. In such case, it becomes vital to monitor the process behavior along with the parent processes that spawn them.

Events generated from Sysmon log is filtered by modifying the XML configuration file. A simple XML configuration file is illustrated below in the figure 4.

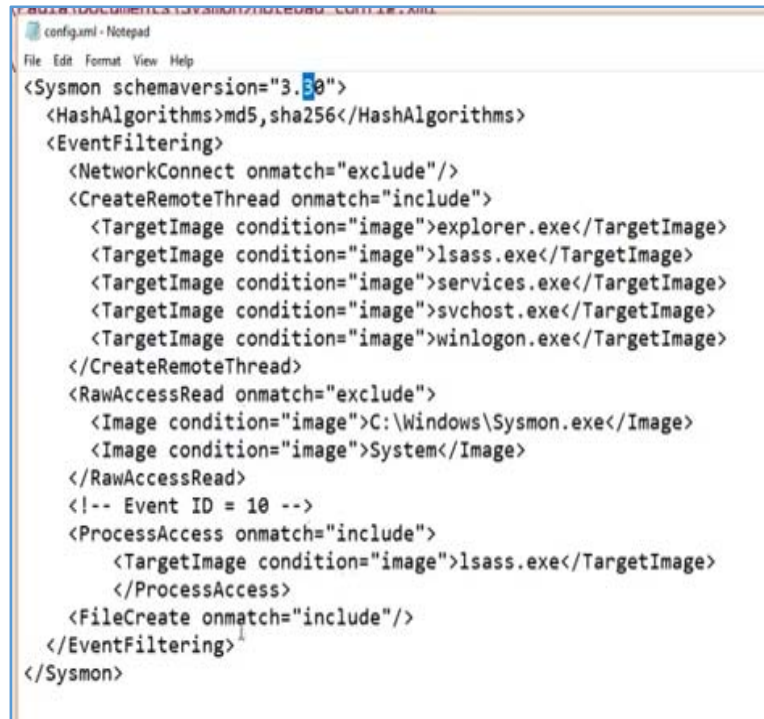


Fig.4. Basic XML Configuration file of Sysmon

Every event has a filter tag like shown in figure 4. The filter tags appear in the EventFiltering node in the configuration file. In this project, several events like Software Protection Service, Windows Update events and many normal events are eliminated from the Event log in order to filter output of host machine and reduce the amount of log to collect.

### Step 3: Winlogbeat and data shipping to Logstash

Winlogbeat ships event logs from Client VM to Logstash on Server VM. Winlogbeat is installed as a Windows service in Client VM. Winlogbeat filters the events according to setting of the configuration file. Beats configuration files are code in YAML. YAML is a file format commonly in use other than XML and JSON. Perhaps, YAML is easy to read and write. All YAML files includes dictionary which contains name:value pairs.

### 4.2. Server Side Implementation

In virtual environment, the server is implemented as a single node with ELK stack. The suspicious event log from the client is forwarded to Logstash present in Server. Setting virtual server node to hunt threats requires the following.

- Set the JAVA Environment
- Install ELK stack on Server
- Use NSSM service installer to install Elastic Search, Logstash and Kibana as services
- Set rules to filter and hunt threats

Server Side Implementation steps include the following

- 1) Install ELK stack on server VM
- 2) Set the stash in Logstash to forward the event log to Elasticsearch
- 3) Inject threats into client VM and stream log in to Logstash and analyze
- 4) Set the detection rules in Elasticsearch to hunt threats

### Step 1: Install ELK stack on server VM

ELK server requires Java for installation. The ELK stack is installed as service on the server VM.

### Step 2. Set the stash in Logstash to forward the event log to Elasticsearch

Event logs from Sysmon are fed to Logstash. Logstash parses each event and identifies the named fields that we intend to, builds the structure and transforms to common format. Filebeat is installed and started to open the event logs in Logstash. Further it is tailed to Kibana. The search bar can be used in default filtering or set criteria for filtering. The service, applications, criteria set, host and data center can be searched.

The Logstash content is in json file format. The transfer of event log to elastic search is illustrated in figure 5.

```
# Sample Logstash configuration for creating a simple
# Beats -> Logstash -> Elasticsearch pipeline.

input {
  beats {
    port => 5044
    add_filebeat => {"[@metadata][source]" => "winlogbeat"}
    ssl => false
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    #user => "elastic"
    #password => "changeme"
  }
}
```

Fig.5. Configuring of stash in Logstash to pull event log data of interest

Kibana is installed as a visualization tool. Kibana comes along with the ELK stack. Filebeat service is started to feed the elastic search results to Kibana. The overall process from pulling event log to display in kibana is shown in figure 6.



Fig. 6. Overview of ELK process on server VM

### Step 3. Inject threats into client VM and stream log in to Logstash and analyze

Detection rules can be implemented on Elasticsearch in two methods. They are

1. Based on characteristics of known threats
2. Based on rule for new kind of threats based on their behavioral analysis

In this project, threat actor injects couple of known threats and unknown threats in to client (victim) from another client (threat actor) as shown in figure 7.

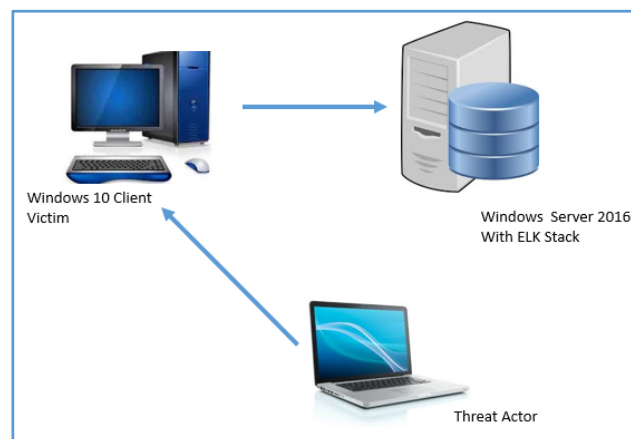


Fig.7. Attack Model for Known and Unknown Threats

Three known threats from ATTK adversary repository were injected in to the client (victim). They are

**Threat 1.** Deletion of files by escalating rights by remote access: Using NMAP the threat actor scans the ports open on client (victim). Access the victim through vulnerable port. Once access is gained, the attacker access victim's files and folders, and deletes them.

**Threat 2. Remote** registry access to create or delete files in the registry of client's (victim) machine. The threat actor uses NMAP to access registry of client from a remote desktop and gets the credentials. Then, he accesses the registry to create or delete files in the registry without the knowledge of the client.

**Threat 3.**Reaver: Using NMAP the threat actor scans the ports open on client (victim). Access the victim through vulnerable port. Once access is gained, the attacker injects Reaver into the victim machine to induce five poison movements. (i.e.) to obfuscate control panel.

**Step 4.** Set the detection rules in Elasticsearch to hunt threats

Based on the output of Elasticsearch, the characteristics the threats discussed in step 3 were analyzed and rules were set to capture multiple incidents of both the cases.

Rules to identify the process's behavior and register.

- 1) If user / process try to access files or folders or other system files or memory need to be filtered
- 2) If any user / process try to scan through open ports
- 3) If NMAP entry in the registry
- 4) If lateral movement in the system
- 5) If control panel is accessed and intended to modify

These rules were incorporated in Elasticsearch to hunt for suspicious events and to capture the details of these events or processes.

The outcome of this ATHS is evaluated by

- Whether the threats injected are identified
- Whether the details i.e. evidences of the threats injected are logged appropriately and filtered appropriately
- Whether alert message as "Threat" is raised to the administrators.

For evaluation and to test the different threat cases the following ips have been used in the simulated environment.

Victim IP: 192.168.1.15

Threat Actor IP: 192.168.1.18

Monitoring Server IP: 192.168.1.16

#### Test Case 1:

**Threat 1.** Deletion of files by escalating rights by remote access: Using NMAP the threat actor scans the ports open on client (victim). Access the victim through vulnerable port. Once access is gained, the attacker access victim's files and folders, and deletes them.

**Result:** The threat actor accessing the victim's files and folders on shared access. The files are deleted in the victim's machine of IP:192.168.1.15. The event with id 5145 is performing the deletion of shared files and the threat alert message with threat\_id 108. The result is shown in figure 8 and alert raised is shown in figure 9. The access to the shared files is checked and deleted further.

```

▶ April 7th 2019, 13:20:22.383  opcode: Info @timestamp: April 7th 2019, 13:20:22.383
                                event_id: 5,145 task: Detailed File Share thread_id: 108
                                message: A network share object was checked to see whether
                                client can be granted desired access. Subject: Security ID: S-
                                1-5-21-2997953885-3869232823-311617462-500 Account Name:

▶ April 7th 2019, 13:20:22.382  event_id: 5,145 @timestamp: April 7th 2019, 13:20:22.382
                                opcode: Info task: Detailed File Share thread_id: 108
                                message: A network share object was checked to see whether
                                client can be granted desired access. Subject: Security ID: S-
                                1-5-21-2997953885-3869232823-311617462-500 Account Name:

▶ April 7th 2019, 13:20:22.381  event_id: 5,145 @timestamp: April 7th 2019, 13:20:22.381
                                opcode: Info task: Detailed File Share thread_id: 108
                                message: A network share object was checked to see whether
                                client can be granted desired access. Subject: Security ID: S-
                                1-5-21-2997953885-3869232823-311617462-500 Account Name:

```

Fig.8Kibana's result with threat hunt

```

▶ April 7th 2019, 13:20:22.383  opcode: Info @timestamp: April 7th 2019, 13:20:22.383
                                event_id: 5,145 task: Detailed File Share thread_id: 108
                                message: A network share object was checked to see whether
                                client can be granted desired access. Subject: Security ID: S-
                                1-5-21-2997953885-3869232823-311617462-500 Account Name:

```

Fig.9.Kibana's result with threat alert message

## Test Case 2:

**Threat 2.** Threat actor gains access to the Registry of the victim's machine and deletes the registry content.

**Result:** In this case, the threat actor scans the open port using NMAP tools. He gains access and tries to delete an application Ex: Notepad from the registry which is shown in figure 10. The event is captured in Sysmon log, falls in the filtered log as per filtering rule in Sysmon. The registry event is captured in Kibana's result with threat\_id 2360 according to the rule which checks for file create or delete in registry is shown in figure 10.

```

▶ April 10th 2019, 09:14:52.958  opcode: Info computer_name: monitoredserver
                                thread_id: 2,360 @timestamp: April 10th 2019, 09:14:52.958
                                type: wineventlog level: Information task: Registry object
                                added or deleted (rule: RegistryEvent) record_number: 47828
                                version: 2 host.os.version: 10.0 host.os.platform: windows

```

Fig. 10. Kibana's result for Registry Access

## Test Case 3:

**Threat 3.** Injection of Malware. Few malware files were injected in to the victim machine by the threat actor.

**Result:** A simulation was performed where a malware executed on the victim's machine. Malware samples are taken from the internet source. The screenshot shown in figure 11 shows the log of a suspicious activity. It is suspicious because the ip address shown is anonymous and could possibly do some harm. This log will help the administrator to track these kinds of activity.



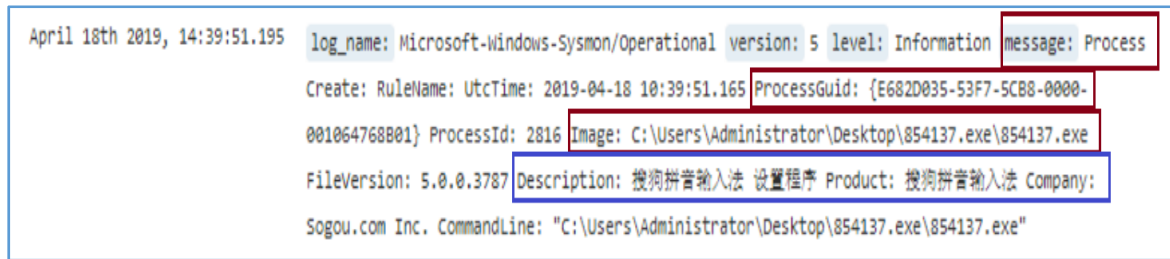


Fig.11.Kibana's Result displaying Malware hunt

In all the above test cases, NMAP is used to detect open ports on remote hosts. It can map the network and enumerate the list of vulnerabilities. Domains and Sub domains can also be queried with massive DNS queries by NMAP to figure out the victim's endpoint vulnerabilities. NMAP scans the victim 192.168.1.15 and the access event is captured by Kibana which is depicted in figure 12.

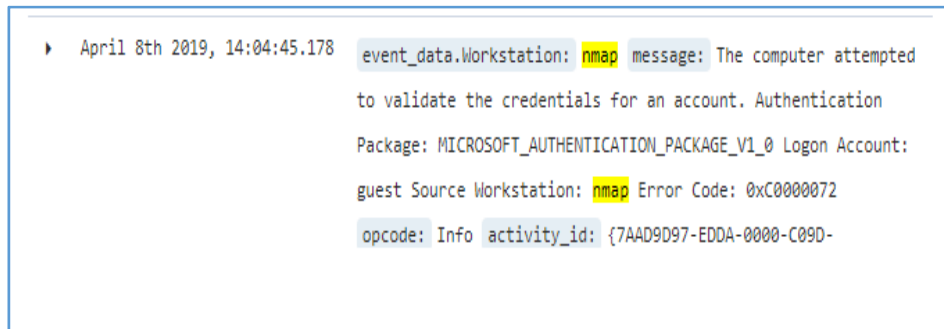


Fig.12.Kibana's Result displaying NMAP access to validate credentials

#### 4. Conclusion

The paper focused on hunting threat automatically by dynamically analyzing the event logs. The complete project was emulated on virtual environment using Oracle VM Virtual Box. An experimental set up of a client VM monitored by a server VM node with ELK stack. Sysmon which resided on client VM extracted all system logs which were filtered for suspicious event incidents. The log was shipped by WinLogbeat in the client VM to Logstash on server VM. The Elastic search analyzes the event logs based on definite rules set to hunt threats. Three threats 1. Remote access and delete files and folder on victim's VM 2. Remote access of victim's VM registry to create and delete files from registry 3. Injection of Malware on victim's VM were simulated. The threats were hunted successfully and displayed in Kibana's result with threat alert message.

In this paper, the behavior of a malicious event were analyzed only based on event ids, parent process that spawned them. Perhaps, the project could expand its dimensions in analyzing and hunting threats based on the factors like persistence in the network and operations carried out by malicious processes.

#### References

- [1] AkshathaSujyothi;Shreenath Acharya. (2017), Dynamic Malware Analysis and Detection in Virtual Environment,IJ. Modern Education and Computer Science, Vol.3, pp. 48-55.
- [2] D.Asir Antony Gnana Singh; E.JebamalarLeavline. (2013), Data Mining In Network Security - Techniques & Tools: A Research Perspective, Journal of Theoretical and Applied Information Technology, Vol. 57(2).
- [3] G.V. Nadiammai; M. Hemalatha.(2014), Effective approach toward Intrusion Detection System using data mining technique's, Egyptian Informatics Journal 15, pp. 37–50.
- [4] Hao Bai, Chang Zhen; Xiao-Chuan Jing; Ning Li; Xiang-yin Wang.(2014), Approach for malware identification using dynamic behavior and outcome triggering, IET Inf. Security, Vol. 8, Issue.2, pp. 140-151.
- [5] Huaglory;Tianfield. (2017), Data Mining Based Cyber-Attack Detection, System Simulation Technology (ISSN1673-1964), Vol. 13, No. 2, Apr. 2017, pp. 90-104.
- [6] IsmahaniIsmail;SulaimanMohdNor;MuhammadNadzirMarsono. (2014), Stateless Malware Packet Detection by Incorporating Naïve Bayes with Known Malware Signatures, Journal of Applied Computational Intelligence and Soft Computing.
- [7] Jabez J; B. Muthukumar. (2015), Intrusion Detection System (IDS): Anomaly Detection using Outlier Detection Approach, International conference on Intelligent Computing, Communication & Convergence (ICCC-2014), science direct, Procedia Computer Science 48, pp. 338 – 346.
- [8] JianfengPeng ;Chuan Feng ; Haiyan Qiao ; Jerzy Rozenblit. (2007),An Event-Driven Architecture for Fine Grained Intrusion Detection and Attack Aftermath Mitigation, 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'07).
- [9] Leandro Velaso;Rik Van Duijn. (2018), Fileless Threats – Analysis and Detection, Whitepaper, Security Reseach Team, dearBytes.
- [10] S Brown; S Carlin; I-Torres Negron. (2017), Next-Generation Defensive Cyber Operations (DCO) platform, Journal of Information Warfare, Vol. 16, Issue.2, pp. 43-55.
- [11] SulijateTantimayteevanich;ChutimaBeokhaimook. (2016), Development Of Centralized Monitoring And Automated Notification System For Ip Network (Sysmon), National And InternationalSripatum University Conference, Pp. 144-148.

- [12] Thanassis Avgerinos; Brent Lim TzeHao; David Brumley.(2011), Automatic Exploit Generation, [www.ndss-symposium.org/ndss2011/aeg-automatic-exploit-generation](http://www.ndss-symposium.org/ndss2011/aeg-automatic-exploit-generation).
- [13] VasileiosMavroeidis;AudunJosang. (2018), Data-driven threat hunting using Sysmon, ACM, ISBN978-1-4503-6361-7/18/03. ICCSP, March 16–19, Guiyang, China <https://doi.org/10.1145/3199478.3199490>
- [14] VasileiosMavroeidis;KamerVishi;AudunJøsang.(2018), A Framework for Data-Driven Physical Security and Insider Threat Detection, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)
- [15] VasileiosMavroeidis; Siri Bromander. (2017), Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence, In Proceedings of the European Intelligence and Security Informatics Conference. IEEE.