Automation of Log Analysis Using the Hunting ELK Stack

Mihail Alexandru STAN
National Institute for Research and Development in Informatics – ICI Bucharest
mihail.stan@ici.ro

Abstract: Computer networks store data about processes, functional parameters and user activity everyday. The information is stored in log files, which have become mandatory in maintaining the security of a system and helping prevent cyber security incidents. Although logs collect useful data, the large amount of information that needs to be processed is a challenge. One of the log management systems is the ELK open source utility stack. This method of automating log analysis incorporates machine learning techniques. Machine learning methods make it possible to identify, with high precision, the differences between normal and abnormal data, and can thus be used to detect different types of network vulnerabilities. This project aims to implement the HELK stack for a computer network in order to streamline log analysis, detect vulnerabilities and lateral movements of malicious software.

Keywords: Log files, ELK stack, Log management, Machine learning, Vulnerabilities detection

INTRODUCTION

Cybersecurity has now become a pressing issue, due to the increasing number and complexity of cyber-attacks on computer systems and computer networks, attacks aimed at accessing, modifying, destroying, unauthorized retrieval of information or disrupting the normal operation of services. Logs are mandatory in the process of developing and maintaining the security of many software systems. The detailed runtime information that is recorded helps the developers and support engineers to monitor systems and analyze abnormal behavior or errors. There are numerous solutions on the market like Log.io, Wazuh, SolarWinds and one of the best being Darktrace. With a very well trained machine learning algorithm and real time vulnerability scanning, Darktrace is one of the most used solutions for a SIEM, but it comes at a very high cost. Apart from the existing competitors, Elastic offers the ELK stack as an open source software, in order to be further developed by the community. For powerful machine learning capabilities, or hosting the services on Elastic Cloud, additional payment will be required, but the endless possibilities of using ELK with a lot of free modules can achieve very similar results like the subscription solutions offer. The goal of this paper is to highlight the

importance of log management tools and the boundless aid that can be provided for securing a network, using the ELK stack, beats and modules for implementation.

LOG ANALYSIS VS. THREAT HUNTING

Log Analysis

Collecting data regarding the current working state of an internal network through the use of log files is a conventional practice nowadays adopted by developers and system engineers in order to help understanding and diminishing the number of menaces that a network faces every day. The rich information and the very high number of equipment that needs to be kept under observation have led to the development of monitoring and diagnosing solutions based on the information extracted from log files. Software tools like this provide usage statistics of the equipment, application security, identification of abnormal performance behavior and error diagnosis. Many recent research and industrial tools have adopted powerful text search and machine learning-based analysis solutions. Due to the unstructured nature of the log, the first crucial step is to parse the log message into structured data for subsequent analysis. In recent years, both academia and industry have conducted extensive research on automatic log parsing, and a series of log parsers have been produced through different technologies in order to be integrated in the large and complex system of log analysis tools.

Threat Hunting

Threat hunting has become one of biggest necessities for any existing organization. The aim of threat hunting is to reduce the time between a cyber-security breach and its discovery. Shortening that time reflects on the budget spent for the remediation of the existing problem. Unfortunately, a large number of organizations are yet to implement hunting teams with specific software tools to ease their work. A recent study revealed that 20% of reported breaches are detected by an external entity (Brent, David, 1997). In order to reduce the overall cost of security while keeping an organization's intranet network secure the hunting team must assume that breaches have already occurred and try to hunt them in order to keep up with the cyber security latest threats (Brent, David, 1997).. This different approach to cyber security demands creation of different hypothesis and application of this possible scenarios in the internal network of the organization. While log analysis relies on reducing the number of indicators of compromise, threat hunting adopted methods like continuous testing of various hypothesis, making a substantial difference regarding network security.

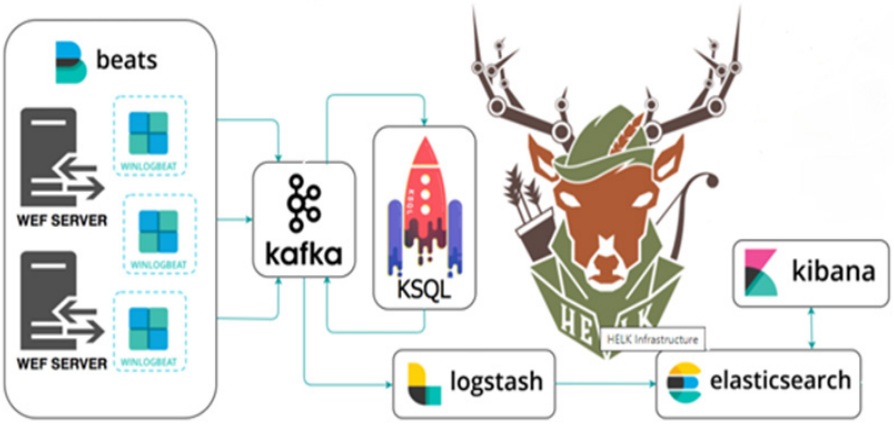

Fig. 1: *Architecture used in the example (Brent, David, 1997).*

HELK OVERVIEW AND TECHNOLOGIES

Log Messages

The first step in automated log analysis is represented by log parsing. This operation thus provides breaking down log messages in order to help train the machine learning oriented log management tool, group objects, filter and generate usage statistics of the equipment. Every log message has a different structure, contains different information and has a different source. Log management tools breakdown unstructured logs and organize them through „shippers".

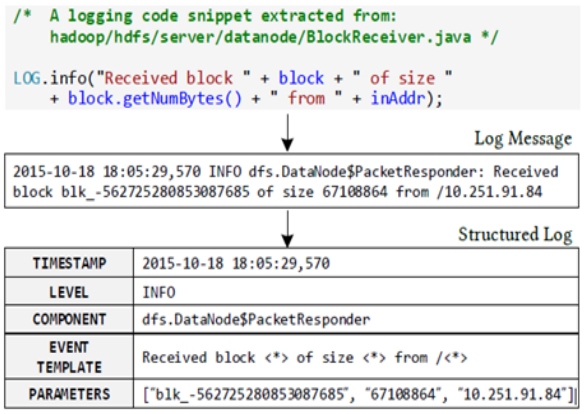

Fig. 2: *Log message example*
Jieming, Z., Shilin H., Jinyang L., Pinjia H., Qi X., Zibin Z., Michael R. L. (2019)

SHIPPING DATA TO ELK STACK

Software solutions based on the ELK stack use beats which act like „shippers", delivering already structured log information from the equipment to the NoSQL database in order to be structured and analyzed for further refinement. The most popular beats are Winlogbeat, Filebeat and Metricbeat. Every beat acts as a service on the machine or the subnetwork depending on where it was deployed. Beats collect useful log information and ship it to the next hop in the log management infrastructure.

Sysmon is a service installed on the Windows machines of interest that provides crucial information about certain actions in the system like process creation, remote connection, access granted for certain processes and the way of creating them. This information is stored and shipped into the infrastructure for later analysis. For every event on screen alerts can be triggered or graphics created especially designed to recognize malicious activity in the network.

Kafka is an open source software which helps SIEM developers to read and analyze streaming data. Kafka is often used in real-time streaming data architectures to provide real-time analytics. It is fast, durable and it is designed to provide fault-tolerance in case of overflooding the database with information.

Suricata is a very powerful and popular network intrusion detection and prevention system. The IDS is one of the most important barriers inside a computer network. Based on protocol, addresses, mac or payload content, traffic can be dropped or an alert can be triggered.

Elasticsearch, Logstash, Kibana

"Logstash is an 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" (Alex, 2019). Elasticsearch is a NoSQL database that allows you to store, search, and analyze huge volumes of data quickly. "It's able to achieve fast search responses because instead of searching the text directly, it searches an index." "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" NovelVista (2021).

PROOF OF CONCEPT

Architecture of the Simulated Test Network

For the purpose of demonstrating HELK's capabilities, the simulated network is represented by the following virtual machines:

- Windows 10, 4GB RAM
- Ubuntu Focal Fosa 20.04, 4GB RAM
- Debian 10, 64 GB RAM

HELK and Suricata were installed on the Debian VM, due to the high system requirements. The Ubuntu VM has Suricata and Filebeat and Packetbeat installed as services. These services are configured using the guides from the Elastic community. On the Ubuntu machine a web server, a file server and a mail server were aditionally configured in order to simulate a real computer network enviroment.

The Windows VM needed more preparation regarding log shipment. The following policies have to be enabled in order for logs to be recorded: Advanced Audit Policy, Enable Audit Process Creation, Enable Force Audit Policy Subcategory Settings, Module Logging, Enable All Task History. Sysmon64 and Winlogbeat were also mandatory for collecting and shipping logs.

Scenarios

First scenario is represented by assuming that there is already a backdoor into our local network, and the malicious actor can execute remote commands in cmd or PowerShell with Administrator privileges. Several events have been captured during the time of monitoring the network:

- Service creation
- Remote code execution
- Security Events Log removal
- Identification of remote connections

The second scenario consists in running Mimikatz, a malware well known in the community of cyber security, in order to gain privileged access and retrieve sensitive information from the system, such as usernames and passwords. A third scenario was created using the hping3 command in order to simulate a DDOS attack.

For a detection to be successful and the results to be stored in the database we need rules in order to detect certain events in the system. Rules contain certain fields that are mandatory for traffic analysis. Event ids, parent process name, process granted access and even the name or port of the process can be used in order to identify threats.

- Rule for detecting Mimikatz:

```
alert:
    - debug
    description: Detect process LSASS
    filter:
    - query:
       query_string:
          query: (event_id:"10" AND target_process_path.keyword:(*\\lsass.exe) AND process_granted_access_orig:0x1fffff AND process_call_trace.keyword:(*dbghelp.dll* OR *dbgcore.dll*))
    index: logs-endpoint-winevent-sysmon-*
    name: LSASS-Memory-Dump_0
    priority: 2
    type: any
```

- Rule for detecting Events Log removal

```
alert:
    - debug
    description: Deleting'Security' Eventlog
    filter:
    - query:
       query_string:
          query: (event_id:(„517" OR „1102"))
    index: logs-endpoint-winevent-security-*
    name: Security-Eventlog-Cleared_0
    priority: 2
    type: any
```

- Rule for detecting rare children created by other services, not from a trusted process path

```
alert:
        - debug
        description: Detecting rare childs from services.exe
        filter:
        - query:
            query_string:
                query: (event_id:1 AND process_parent_name:"services.exe" AND NOT (process_path:"\\windows\\system32" OR process_path:"\\program\ files\\windows\ defender\\mpcmdrun.exe" OR „\\program\ files\ \(x86\)\\google\\update\\googleupdate.exe" OR „c:\\windows\\servicing"))
        index: logs-endpoint-winevent-sysmon-*
        name: Windows-services-rare-child_0
        priority: 2
        timestamp_field: etl_processed_time
        type: any
```

RESULTS

At the end of every scenario, the events were successfully captured and could be analyzed in Kibana, the web interface for monitoring logs. What can be visualised in the Kibana web interface depends on the rules and triggers that have been configured on the HELK server. Events like administrator commands, malicious process creation or execution and abnormal traffic activity were identified. Using the Machine Learning tab and providing an index pattern, ELK offers the capabilities of learning what normal activity inside the network looks like and can trigger alerts for suspicious events. Users can create personalized queries and dashboards of preference in order to ease the process of monitoring relevant network traffic. More detailed information about every event can be found in the "Discovery" tab. Results provide a basis for monitoring network security events and training hunting teams for future security breaches.

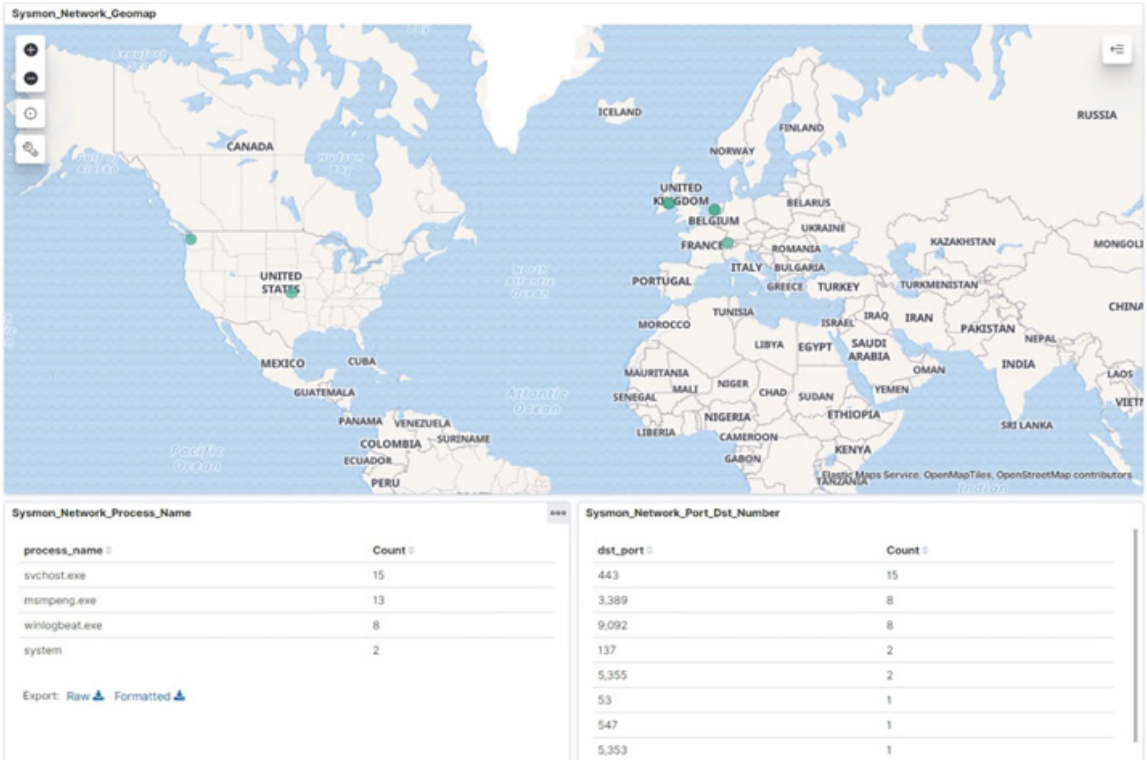

Fig. 3: *Kibana Dashboard*

CONCLUSIONS

Every event in the system, every command and every action leaves behind a trace marked by a log message. It is up to the users of this solution what information is more important, what to be transformed in alerts or be registered on a graphical representation. Using the HELK stack not only attacks can be simulated and analyzed for training purposes but it is also a viable tool for managing a network's log files with the purpose of improving threat hunting capabilities. At the moment HELK is still developing, having a very big and creative open source community behind it. The security of every system depends on the creativity and curiosity to create and exploit vulnerabilities from inside the network in order to learn and know how to protect against them. Future research should consider the use of traffic anomalies such as packet flow, equipment usage and usual working hours for certain devices, in order develop more elaborated rules with the purpose of identifying cyber-attacks.

AKNOWLEDGEMENTS

The presentation of this project was possible due to the bachelor project "System for Automation of Log Analysis Inside Computer Networks".

REFERENCE LIST

Alex, R. (2019) Centralized logs with Elastic stack and Apache Kafka, retrieved from: https://medium.com/inside-freenow/centralized-logs-with-elastic-stack-and-apache-kafka-7db576044fe7

Brent M., David F. (1997) CyberEdge Group, The Elastic Guide to Threat Hunting. retrieved from: https://www.elastic.co/pdf/elastic-guide-to-threat-hunting

Cyb3rWard0g (2020), retrieved from: https://github.com/Cyb3rWard0g/HELK

Cyb3rWard0g (2018), retrieved from: https://cyberwardog.blogspot.com/2018/04/welcome-to-helk-enabling-advanced_9.html

Das, R., Johnson, G. (2020). Network Security

Jieming, Z., Shilin H., Jinyang L., Pinjia H., Qi X., Zibin Z., Michael R. L. (2019) Tools and Benchmarks for Automated Log Parsing, retrieved from: https://arxiv.org/abs/1811.03509

Maheshwaram, S. (2018). A Study on Security Information and Event Management (SIEM). 5. 705-708.

NovelVista (2021) An ELK Stack Tutorial, retrieved from: https://www.novelvista.com/blogs/devops/elk-stack-tutorial

Ostrovsky, D., Rodenski, Y. (2014). ElasticSearch Integration. 10.1007/978-1-4302-6614-3_8.

Sachdeva, G. (2017). Introduction to the ELK Stack.