

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Information Technologies

Fernando Bauzá Sainz de Baranda - 184585IVCM

**HOW TO MESS WITH LOG COLLECTORS  
AND ANALYSE THEIR RESPONSE IN  
MICROSOFT NETWORKS WITH AN  
EXAMPLE OF THE ELK STACK.**

Master's Thesis

Supervisor: Toomas Lepik  
MSc Cybersecurity

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Fernando Bauzá Sainz de Baranda - 184585IVCM

**LOGIKOGUJATE TÖÖ HÄIRIMINE NING  
NENDE REAGEERINGU ANALÜÜS  
MICROSOFTI VÕRKUDES ELK-PINU  
NÄITEL.**

Magistritöö

Juhendaja: Toomas Lepik  
MSc Cybersecurity

Tallinn 2020

## **Author's declaration of originality**

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Fernando Bauzá Sainz de Baranda

20.04.2020

## **Abstract**

In internal networks where event collection has been centralized to facilitate their monitoring, it is essential to detect alterations in the integrity of the events. Log collectors gather these events from different system sources and represent a target for attackers seeking to manipulate the logs. This thesis seeks to analyse the behaviour of log collectors in Microsoft networks, studying the traces left by these attacks and the situation in which the manipulated events arrive. For this purpose, a study has been conducted applying attacks on the different vulnerable points of the event flow: the source of the data, the transit and the log collectors themselves, using the Winlogbeat and Filebeat log collectors on an internal network that simulates a small company architecture. The results have demonstrated the capacity to alter the events at the source of the data and in the attacks directed at the log collectors themselves, reaching the Kibana service with an appearance very similar to normal events. The result of the attacks as well as the traces left by them provide information from a focused scope that can help analysts detect these attack patterns and protect systems against this kind of threats.

Keywords: Log collector, ELK Stack, log manipulation.

This thesis is written in English and is 59 pages long, including 7 chapters, 22 figures and 2 tables.

## **Annotatsioon**

### **Logikogujate töö häirimine ning nende reageeringu analüüs Microsofti võrkudes ELK-pinu näitel**

Tervikluse rikke tuvastamine on üks olulisemid probleeme arvutivõrkudes kus on kasutusel tsentraliseeritud logide kogumine. Erinevatest allikates saabuavad logid võivad olla rünete sihtmärgiks mille eesmärk on manipuleerida keskselt kokku kogutavat logi. Käesoleva lõputöö analüüsib logikogujate käitumist Microsoft võrkudes, uurides jälgi mis on jäetud rünnete poolt ja olukordi millal manipuleeritud sündmused saabuavad. Selle tarbeks vaadeldi erinevaid turvanõrkusi logi kogumise protsessis st. logi allikat, logide edastamist ja logi koguja enda parameetreid kasutades logi kogumiseks ja edastamiseks Winlogbeat ja Filebeat logi kogujaid sela juures simileerides väkeette võtte võrgu arhidektuuri. Tulemused näitavad, et sündmuste muutmine on võimalik nii lago allikas kui ka logi kogujatele rünnetes mille tulemusana andmed mis jõuavad kesksesse andmebaasi ning mida näidatakse läbi Kibana teenuse on tavaliste sündmustega ära vahetamsieni sarnased. Välja toodud rüünnete tulemid ja need poolt jäätud jäljed anavad ülevaate analüütikule kuidas neid tündemustreid tuvastada ja süsteeme selliste rünnete vastu kaitsta.

**Märksõnad:** Logikogujate, ELK-pinu, logiga manipuleerimine

Lõputöö on kirjutatud Inglise keeles ning sisaldab teksti 59 leheküljel, 7 peatükki, 22 joonist, 2 tabelit.

## **Acknowledgements**

First, I would like to thank my supervisor, Toomas Lepik, for his patience and recommendations during the realization of this thesis that have led me to sort out my ideas and improve my work.

Secondly, I would like to thank, in this difficult world context under which we have worked, the support of friends and family, especially my father, who has been a pillar in the most complicated moments.

To all of you, thank you very much.

## List of abbreviations and terms

RCE	Remote Code Execution
ID	Identifier
SCADA	Supervisory Control And Data Acquisition
ELK	(Elasticsearch, Logstash and Kibana) referring to Elastic main products
UDP	User Datagram Protocol
IP	Internet Protocol
TCP	Transmission Control Protocol
RELP	Reliable Event Logging Protocol
TLS	Transport Layer Security
SSL	Secure Sockets Layer
FIFO	First In First Out
WEF	Windows Event Forwarder
NTLM	New Technology LAN Manager
MB	Megabyte
MiB	Mebibyte
KiB	Kibibyte
JSON	JavaScript Object Notation
OS	Open Source
NOS	Not Open Source
HTTPS	Hypertext Transfer Protocol Secure
RAM	Random Access Memory
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
AD	Active Directory
NAT	Network Address Translation
HTTP	Hypertext Transfer Protocol

## Table of contents

List of figures .....	9
List of tables .....	10
1 Introduction .....	11
2 Previous Works .....	14
3 Log Collectors .....	17
3.1 Vulnerable Points .....	21
4 Environment Set-Up .....	24
4.1 Server configuration and installation .....	24
4.2 Log storage and Log shipping .....	28
4.2.1 Elasticsearch .....	29
4.2.2 Winlogbeat .....	30
4.2.3 Filebeat .....	31
5 Experimental Research .....	33
5.1 Evaluation on Winlogbeat .....	34
5.2 Evaluation on Filebeat .....	40
5.3 Discussion and Validation .....	44
6 Prevention and Detection .....	46
7 Conclusion .....	50
References .....	53
Appendix 1 – PowerShell command for log injection on Winlogbeat Transit .....	56
Appendix 2 – PowerShell command for log injection on Filebeat Transit .....	58



## List of figures

Figure 1. Log shipping data flow of Windows event logs 4624 (successful login) or 4625 (failed login). .....	22
Figure 2. Vulnerable points in the log collector login data flow. ....	23
Figure 3. Server architecture. ....	25
Figure 4. Log transfer esquema. ....	28
Figure 5. Elasticsearch.....	29
Figure 6. Winlogbeat configuration file (winlogbeat.yml).....	30
Figure 7. DNS debugging log files.....	31
Figure 8. Configuration of Filebeat (filebeat.yml). ....	32
Figure 9. Log flow architecture. ....	32
Figure 10. Log attack points.....	34
Figure 11. Normal user2 logon events.....	35
Figure 12. Visualization status after log cleared and process stoped. ....	37
Figure 13. Fragment of standard running service information log in Kibana. ....	37
Figure 14. Generated System 7036 event.....	38
Figure 15. Index list from Elasticsearch.....	39
Figure 16. Successful response from Elasticsearch after POST insertion on Winlogbeat index. ....	39
Figure 17. Time jump after altering Winlogbeat configuration. ....	40
Figure 18. Filebeat normal log flow. ....	41
Figure 19. Results after log insertion at Filebeat's events source.....	42
Figure 20. Successful response from Elasticsearch after POST insertion on Filebeat index. ....	42
Figure 21. Effects of surpassing maximum message size on Filebeat. ....	43
Figure 22. Path change in Filebeat. ....	43

## **List of tables**

Table 1. Characteristics of log collectors. ....	21
Table 2. Summary of the elements of the network.....	27

# 1 Introduction

Within the whole business framework and its computer networks, cybersecurity has been growing in importance gradually intending to protect the security of their data and users. The different security systems used to protect us from computer threats have been evolving and improving at the same time as its importance was growing, constantly raising security patches and new tools. However, the threats have also been progressing, looking for those vulnerable points to be able to access the computer systems and affect, in one way or another, this environment that we are trying to protect. What we want to establish with this paragraph is that you cannot be prepared for everything, every patch released by an operating system has in response a new exploit or a new path to access that has not been covered. This is where the importance lies of not only preventing attacks but also detecting them when they occur, knowing where they come from, what elements they have accessed or how they have done it. One practice that helps in this field is log monitoring.

Monitoring the different system events helps to detect abnormal changes that may have occurred both in the internal network and in the system among other components. For example, if a subject with malicious intentions were to establish a remote connection to one of the computers on the internal network, he would leave a trace on the system security events. This event can be studied by the cybersecurity analyst, detecting the threat and studying the reasons that led to it in order to prevent a future attack with the same method. However, attackers also make use of a technique that takes advantage of this dependence on logs to deceive the analyst or cover their tracks after their attack. Log manipulation attacks, understood as log injection, tampering or forging, are attacks that consist of the creation, editing or deletion of events with the aim of tricking the analyst and hide the trail of an attack or executing it [1]. This causes a loss of confidence in the contaminated events and may lose their ability for forensic use.

In Linux environments, this method is more documented, especially when it comes to injecting commands into the system logs to execute them remotely (RCE) [2]. Other

ideas and techniques to protect against log forgery and to deceive the user can also be found in different blogs. This is probably because Linux platform has been more thoroughly researched not only in academic papers but also in web blogs and grey sources. This results in the difficulty that **you can find far fewer articles describing these situations in Windows environments**. But even though it is not so documented, log tampering attacks are not so rare in this operating system.

In Microsoft networks of small or large companies where events are centralized, it is crucial to detect a loss of integrity in the logs that could mean a covert attack. Here lies the importance of analysing the behaviour of log collectors, which are responsible for transporting events from their different sources to the storage or control centre, when faced with log manipulation attacks. The different characteristics of the log collectors on the market can lead to different responses to attacks of this type. Studying their behaviour in these situations within the Microsoft environment can provide new information for analysts seeking to detect possible manipulations that might occur on their network.

Throughout this project we seek to analyse different ways of attacking the log collection systems to exploit various situations in the log collectors. This tries to offer different scenarios in which the attacker tries to cover the traces of his intrusion, manipulate the user in such a way that he believes that the target that the attack seems to be looking for, really is a different one or cause a malfunctioning of the collectors. The final objective of this research is to offer an analysis of their response, under the hypothesis that attacks at different points in the process carried out by log collectors on a Microsoft network allow a manipulation of the final state of the event at its point of display. The aim was to find out what kind of traces these attacks leave and the possibility that they may fool any filters an analyst may have to detect malicious attempts on the network. Another detail to observe is the response of log collectors to attacks that force the limits of their capacity.

The methodology used for this purpose has been an **experimental research**, trying to identify the vulnerable areas in the data flow that directly or indirectly influence the log collector's target. This scenario has been presented in a test environment that simulates a small company's internal network. This makes possible to get a more precise scope, avoiding the massive data generated in bigger networks and recreate situations and

events similar to those that could be found in a real situation. We also analysed different characteristics of the log collectors, focusing on some of those that make up the ELK Stack. The test environment has been built on a single computer, where different virtual machines are created to simulate this approach. Finally, to validate the results, the inserted logs have been used together with events of a normal flow, to determine the possibility that these can deceive the filters and the user.

Since the objective is to study the behaviour of the log collectors and not to evaluate the security components of the operating system, different protection functions that in a real situation should be active have been deactivated. This section serves to establish the assumption that this security barrier has been breached and therefore the attacker has already achieved a series of permissions that have allowed him to access the system before manipulating the events.

This thesis has been divided into the following chapters in order to detail the research process carried out. After this first introductory section, section 2 delves into the literary background and the different works related to the topic of this thesis, as well as the main references to carry it out. Next, section 3 details the theoretical part and the functioning of log collectors as well as their characteristics and possible vulnerabilities. The next chapter describes the setting up of the test environment, together with the configuration of the different servers prior to the experimentation process described in section 5, where, in addition to carrying out the tests and attacks on the log collectors, the results obtained are validated and discussed. Section 6 briefly describes some protection methods that stranded out after the experimentation and other recommendations found on other researches. Lastly, chapter 7 includes the conclusions obtained from the results of this research.

## 2 Previous Works

Before the realization of this project, several documents and researches have been studied to provide a guide to the steps to follow during the research. The information obtained from these sources was intended to shed more light on the use of logs as a threat detection tool and the participation of log collectors in this task, as well as other possible experiments that would further the study of log manipulation. Also, different works have been collected about different methods of prevention against event manipulation attacks and how these practices provide greater security in these situations.

For the documentation of the utilities and problems of log management, the *Guide to Computer Security Log Management* [3] provided certain guidelines for the construction of the architecture for the transport and storage of events. This guide seeks to provide a greater understanding of computer security log management from a practical and realistic point of view. The sections covered in this document deal with the log management infrastructure, the planning and the operational process. Among the information provided is chapter 2.3.1, which summarizes the problems faced by event analysis at the organizational level due to its many sources and formats. *Security Log Management* [4] also reviews some aspects related to security event management. Here some problems related to security event monitoring are described again and some solutions are briefly developed.

Different events generated by Windows can point out different types of threats or attacks to the system. The document *Spotting the Adversary with Windows Event Log Monitoring* [5] was prepared to assist the United States Government and Department of Defence administrators in setting up a centralized logging system. The guide classifies different groupings of Windows events by ID that can be linked to certain patterns and situations on the system. Although focused on built-in Windows tools, the source allows for the identification of collected events and their merging within a security context. Another document related to Windows threat detection that has provided information

about events and their correlation with different patterns is *Detecting Security Incidents Using Windows Workstation Event Logs* [6]. In this case, the report shows different techniques to detect some attacks directed to the user. It is interesting to relate, in some cases, individual events to be considered when detecting these abuses.

In the creation of an architecture for the transmission of logs and which software to use as a basis for the project, research has been studied such as *Scada Statistics Monitoring Using the Elastic Stack* [7] where Elastic Stack (products of Elasticsearch) is used to monitor SCADA statistics in which it is concluded that, although with superficial data, the results have been easily managed by this tool package. Another example analysed for the study of the characteristics of ELK Stack has been *Performance of ELK Stack and Commercial System in Security Log Analysis* [8] where the use of ELK Stack is recommended to build a security log analysis system for medium or small companies in a comparison between commercial products and open-source software. In *Log monitoring and analysis with rsyslog and Splunk* [9] another architecture for log analysis is studied using a system based on rsyslog and Splunk that shows another example of a centralized event system while comparing different features. Some blogs such as *Windows Event Forwarding for Network Defence* [10] were also explored to complement the information available in academic papers regarding the characteristics of different log collectors. In this case, I studied the implementation of the WEF together with its characteristics and limitations. These studies provided multiple features to be considered during the development of the test environment and the assessment of different architectures focused on the centralization of events based on the efficiency and limitations offered by these models.

The documents related to the evaluation of event handling techniques are not very numerous. Although more information can indeed be found in different blogs of the network such as *Abusing Elastic's Beats to Avoid Detection and Manipulate Logging* [11], where some measures that can be practised to abuse ELK Stack are discussed, the main idea to classify and perform different attacks to the event flow was provided by the *Logging and log management guide* [12]. The guide in general deals with various concepts of log management and how to manage events. From this guide, we have focused on chapter 17: Attacks Against Logging Systems which develops further the information you were looking for. It details different attacks in a general way applied to the parameters of confidentiality, integrity and availability of logs.

Other research consulted to gather more information in the context of protecting the integrity of events in log management systems is related to log protection and security measures. The first of these, *Efficient Record-Level Keyless Signatures for Audit Logs* [13], proposes a signature system for the events in order to provide a method of verification and thus confirm that the integrity of the log has been maintained. Next, several papers propose security systems to protect the integrity of the events based on blockchain. *A Prototype Evaluation of a Tamper-resistant High Performance Blockchain-based Transaction Log for a Distributed Database* [14], shows a prototype in a decentralized event system that concludes with results that prove to be effective and maintain event integrity. *Logchain: Blockchain-assisted Log Storage* [15], presents another type of blockchain system applied to the Cloud and that works with a hierarchical ledger. *SDNLog-Foren: Ensuring the Integrity and Tamper Resistance of Log Files for SDN Forensics using Blockchain* [16], on the other hand, describes a mechanism to secure sensitive events for forensic purposes using blockchain technology by preventing attacks on these events. Finally, *Efficient Tamper-Evident logging of Distributed Systems via Concurrent Authenticated Tree* [17], shows a model that avoids log tampering attacks through a concurrent authenticated tree that works efficiently.

The evaluation and search of the multiple documents related to the topic of this thesis make evident the lack of research regarding the behaviour of log collectors when faced with event manipulation. Although some references mention different methods of attack [12] or detection of event manipulation [18], they are mostly oriented to Linux systems, so they are not methods directly applicable to Microsoft networks without a correct orientation to their format and operation. In addition, apart from direct attacks on events and their construction, it is necessary to take into account the interlocution made by the log collectors up to the storage server. The influence of the log collectors could alter the way in which the attacks affect the event and the final display of the system state, altering the different indicators and alarms that can detect these attacks by the analysts and the different means that can be used to prevent them. This thesis aims to fill a gap by providing an analysis of the behaviour of ELK Stack log collectors in a Microsoft network and how they affect log poisoning attacks during the data flow through their characteristics and configuration.



### 3 Log Collectors

The importance of event monitoring in detecting threats to the system has now been established. However, in the face of large computer networks with multiple data sources, this task scales exponentially. As stated in *Graylog* [19], centralization makes it difficult for criminals to manipulate the events and records that remain after their actions. By having events stored in one place, it provides a clearer picture of the system situation, giving analysts more data to detect unusual patterns or activities. To transport the events generated by each source in the system to the centralized storage services, log shippers or log collectors are used.

When managing the different logs produced by the server, several problems complicate the collection of this data [3]:

- Multiple data sources: The different logs are located in multiple folders and directories. Most Windows events are recorded in its main directory, accessible from Event Viewer
- Content: Not only the format of the files where the logs are registered can vary and even make them unreadable for certain log shippers, but also the content that each program wants to register in the events changes. In general, the data sources record the most important content and although some elements can be maintained such as the timestamp, this can be recorded in multiple ways, mentioning seconds, or varying the order of the date.
- Timestamps: The logs register the date and internal time of the application or the equipment. If for any reason these dates are imprecise between several computers such as the server and the client of our internal network this could make difficult its analysis.
- Format: Each source uses its methods to record the events. From the elements of separation to the nomenclature of the applications can vary from one to another.

Some are designed to be readable by users while others are not. There are many ways to sequence these variations, so it is difficult to establish a pattern.

To manage these problems different types of log collectors will transmit the events to the storage server. The choice of the log collector depends on the user's needs in terms of the data sources from which the information is to be extracted for analysis, the technical characteristics of the budget available for the application. In the development of this research, different tools have been studied to fulfil this function taking into account several characteristics that can influence its behaviour when handling log manipulation attacks.

The first point of differentiation between log collectors is the differentiation between Open Source programs and commercial tools. The advantages and disadvantages of both depend on the user's needs. Although the payment tools are not necessarily better than the Open Source examples, they generally include more complete solutions or with greater capacity to process and extract the events from the different sources of the equipment. This may be important when thinking about log manipulation attacks, as greater capacity or better performance can more effectively prevent different attacks.

Another interesting piece of information to consider is the protocol used to transfer the logs to their next destination. As mentioned in some references on Linux systems [8], in messages transported via UDP there is no authentication system for IP addresses in the header, so it would not be possible to check who the sender is. If the server cannot distinguish a malicious source from the log collector, an attacker could use this vulnerability to transfer fraudulent logs or overload the recipient.

The use of different encryption systems is also a significant feature in log collection tools. Encryption or protection of data during transmission can prevent attacks on the integrity and confidentiality of events. Either by carrying out a fraudulent transmission of events as explained in the previous paragraph or by intercepting the transmissions, it can be preventive to establish an encryption system that prevents the attacker from reading or accessing the events themselves and thus avoiding their manipulation.

Finally, another important piece of information within log shippers is the maximum length of the message they carry and the capacity of the buffer to transmit the data at once. The length of the message can affect how events are transferred to the storage

server. This can happen in a way that the event is divided into two parts or if it is discarded for not complying with the parameters established by the collector. If an event is discarded, an attacker could generate a log long enough not to be detected by the system, or if the event is split up it could simulate that they are two completely different events. In the case of buffer capacity, overloading the collector's event transfer rate may allow the attacker to gain some time to perform his malicious actions and erase his trail before the event store can be notified.

Following these parameters, the following log collectors and their different characteristics have been analysed when performing their function:

Rsyslog [20] is an Open Source log shipper with a modular design capable of receiving information from different sources and transmitting it to a storage service. Regarding the transfer protocols used by Rsyslog, the documentation [21] specifies the capability of this tool to perform data transfers via UDP and TCP protocols. In the case of the TCP protocol, Rsyslog provides a more reliable addition through the Reliable Event Logging Protocol (RELP). The encryption protocols that protect the transfers are based here on the TLS/SSL protocols [9]. This requires the sender to authenticate himself to the receiver and the receiver to the sender. This mutual authentication prevents man-in-the-middle attacks. The default message size accepted by Rsyslog is 8k [22] within the *\$MaxMessageSize* variable. The document referring to this measure does not suggest lowering it to 1k as it could cause interoperability problems. Regarding the buffer capacity, in this case, there are two parameters to take into account. Rsyslog uses a queue system where if a high occupancy rate is reached it starts discarding messages to make room for new ones. The default size of this queue is 1000 messages (*queue.size*) and the percentage from which it starts discarding is 80% (*queue.discardMark*) [23].

Syslog-ng is very similar to rsyslog in many ways. Both are Open Source programs that allow the transport of logs without using too many resources and that allow multiple sources of events. As far as encryption is concerned, syslog.ng also allows encryption based on TLS/SSL protocols. For the transfer of messages can be configured using the protocols of UDP and TCP, this time without the support of RELP which provided greater security. The messages received by syslog-ng have by default a maximum capacity of 65536 bytes (*log-msg-size()*) including all its structure and individual fields.

The output capacity of the events transmitted by the log collector works with a FIFO protocol whose output capacity is 10000 messages [24].

Another interesting log shipper is the Windows Event Forwarder (WEF). This tool, typical of the Windows operating system, focuses on the transmission of logs concerning system and operational events. It is, therefore, an Open Source program that allows collecting the most important events. The log collector allows the transport through the TCP protocol to the storage servers. In this case, security is not only based, as seen in the previous examples, on the use of encryption by TLS/SSL but also uses Kerberos-based encryption by default and with NTLM (New Technology LAN Manager) in the background. The buffer size varies depending on the channels used. Little information could be extracted about this, but the extracted data points to a default size of 100MB [25].

Winlogbeat and Filebeat are two Open Source log collectors that belong to the Elasticsearch "Beat" family inside the ELK Stack. Winlogbeat is specifically oriented to the collection and transmission of Windows events. It can similarly collect events as WEF does according to the user's configuration and transmit them to the storage system for later analysis. This system uses the TCP transmission protocol to send the logs to their destination [26]. To protect this transfer, TLS/SSL encryption can be enabled in your configuration file [27]. The messages handled by this application can have a maximum size of 1000000 bytes in JSON format. On the other hand, the speed at which they are transferred depends on the destination of the transmission, in case of Elasticsearch it is by default 50 messages per dump. Filebeat, on the other hand, is centred on the events saved in different files of the computer. It is especially effective when transmitting system and application events in a centralized manner. As with Winlogbeat, it uses the TCP transport protocol and a TLS/SSL encryption system. Also, in case you want to transfer to Elasticsearch, it has a default dump capacity of 50 messages per interaction [28]. Events can be received on either TCP or UDP channels, with TCP being the maximum accepted message size of 20MB and 10KiB over UDP.

Splunk can be considered a more complete tool than the previous ones since it offers a filtering and analysis system of the collected events, but it is not Open Source. The logs can be received in Splunk by the TCP or UDP protocols but when transmitting the data it uses the TCP protocol. The size of the buffer is fixed by default to 10MB and it

cannot be less than 1MB. In terms of encryption, this program also makes use of TLS/SSL certification to ensure the integrity of its transmissions [29].

Finally, Datadog is another non-Open Source program that has capabilities to transmit events like the log collectors mentioned above. This application also allows the reception of data by TCP and UDP protocols but in this case, besides the transmission by TCP, Datadog allows the transfer of events by HTTP protocol. Encryption is once again employing TLS/SSL certifications.

As a summary of the data collected for these log shippers can be seen in the following table:

Table 1. Characteristics of log collectors.

<b><u>Log Collectors</u></b>	OpenSource (OS)/NotOpenSource (NOS)	Transport Protocol	Message Size	Encryption	Buffer size
Winlogbeat 7.6	OS	TCP	Default 10000000 bytes	TLS/SSL	50 messages
Filebeat 7.6	OS	TCP	20MiB/10KiB	TLS/SSL	50 messages
Windows Event Forwarding 2019	Built-in	TCP	-	Kerberos NTLM TLS/SSL	100MB
Rsyslog 8.2001.0	OS	TCP/UDP/REL	Default 8K	TLS/SSL	1000 messages
Syslog-ng 3.25	OS	TCP/UDP	Default 65536 bytes	TLS/SSL	10000 messages
Splunk 8	NOS	TCP	-	TLS/SSL	10MB
Datadog v7	NOS	TCP/HTTP	-	TLS/SSL	-

### 3.1 Vulnerable Points

Figure 1 shows the process by which events 4624 or 4625, which represent a user's successful or unsuccessful login [30], are transported to the central storage system. When these events arrive at the Windows storage system, the log collector collects them together with all the new events that have arrived at the source. These events are

transferred to the storage system together with the source details accessed by the log collector [31].

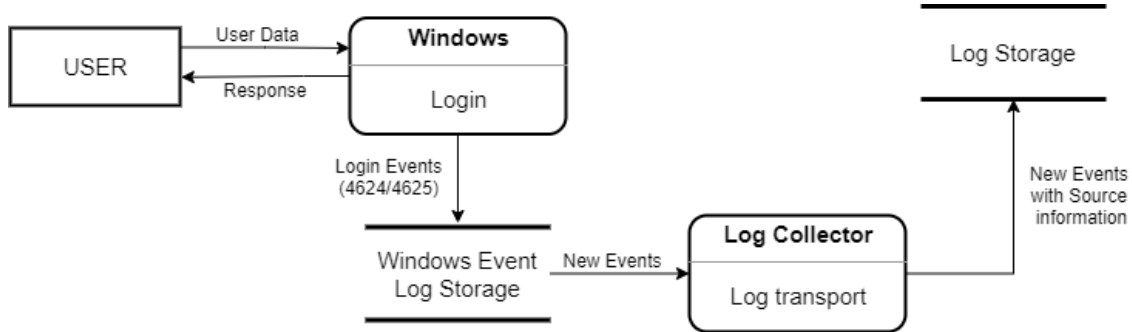


Figure 1. Log shipping data flow of Windows event logs 4624 (successful login) or 4625 (failed login).

There are several points in the flow of these events to the database where the logs may be vulnerable to log manipulation attacks that would affect the log collector's ability to transmit the information correctly. The *Logging and Log Management* guide [12] describes several points to target log attacks to affect the confidentiality, integrity or availability of events. In order to check different situations that directly or indirectly affect the log collector, the following three points will be addressed:

- Attacks at the Source: These are the attacks made to the host that has generated the events or to the application itself. We also include in this category the server folders where the events are stored locally and the Event Viewer service that handles the events of the Microsoft system.
- Attacks in Transit: In this phase, the attacks are directed towards the event flows occurring from the source to the log shippers or from the log shippers to the storage system. Overflow attacks can also be considered and can be grouped in this category.
- Attacks in the Log Collector: This category groups the attacks made directly on the log shipper's system in which it could cause a malfunction or loss of availability.

These will be the categories in which the attacks, applied to the flow seen above, have been classified Figure 2. Vulnerabilities at the source could occur in both user input and Windows storage. The events are also accessible during transport to the log collector and subsequent delivery to the storage system. Finally, an attack concerning the characteristics or configuration of the log collector could also lead to a malfunctioning of the information transmission.

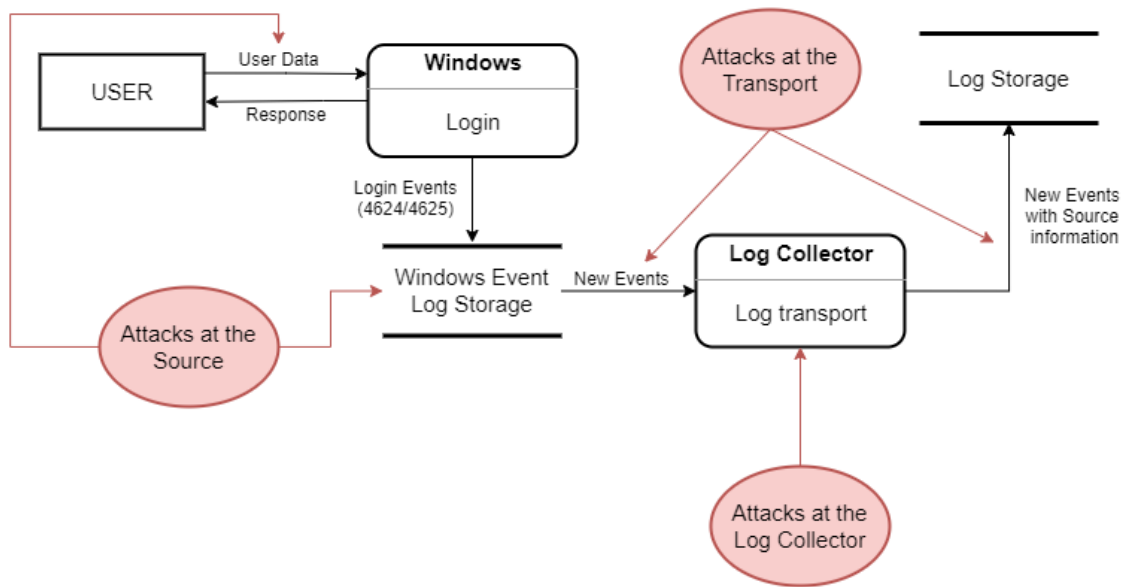


Figure 2. Vulnerable points in the log collector login data flow.

## 4 Environment Set-Up

To carry out this research, there are several systems that must be configured beforehand to obtain better results. In order to collect events in the collector that can be identified with everyday situations in a small company, an environment that simulates an internal network controlled by a server has been provided. This network has been built in a single computer since it is the only hardware available, so some features have been limited to reduce the amount of RAM memory consumed as will be detailed later. The computers that make up the network have been installed in the virtual environment of Oracle VirtualBox version 6.0.18.

In order to explain its setup, this section has been divided into two parts. The first one consists of the processes used to create the server with its different components. The second will focus on the configuration of the log collector and its event transfer systems from the server.

### 4.1 Server configuration and installation

The server will have the function of centralizing the logs and controlling the permissions of the users connected to its internal network. In this thesis, the operating system used to perform this function has been Windows Server 2016 Essentials 64-bit<sup>1</sup> because it is a sufficiently current version of a Windows server and of which there is more information than its latest version of 2019. The following items have been configured to control the different actions taken by users connected to the server:

- DNS Server: It will regulate the resolution of names in the server and therefore through its events we can access information from different searches that requests sent by users.

---

<sup>1</sup> <https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2016> [21st of March 2020]



- DHCP Server: It will allow to assign different IP addresses to the users belonging to your internal network. The logs provided by this service can allow us to identify the different users and the addresses assigned to their computers.
- AD Server: It will serve for the creation of the different user accounts and will register in its events different security functions such as the assignment of permissions, the creation of groups with privileges among other characteristics.

The image of Windows Server 2016 has been mounted in VirtualBox under the name "ThesFerServer" with two network adapters, one NAT to be able to access the Internet from the same machine and to be able to download and install the different components as well as to upgrade the computer, and another single-host to allow the creation of the DHCP server as can be seen in Figure 3

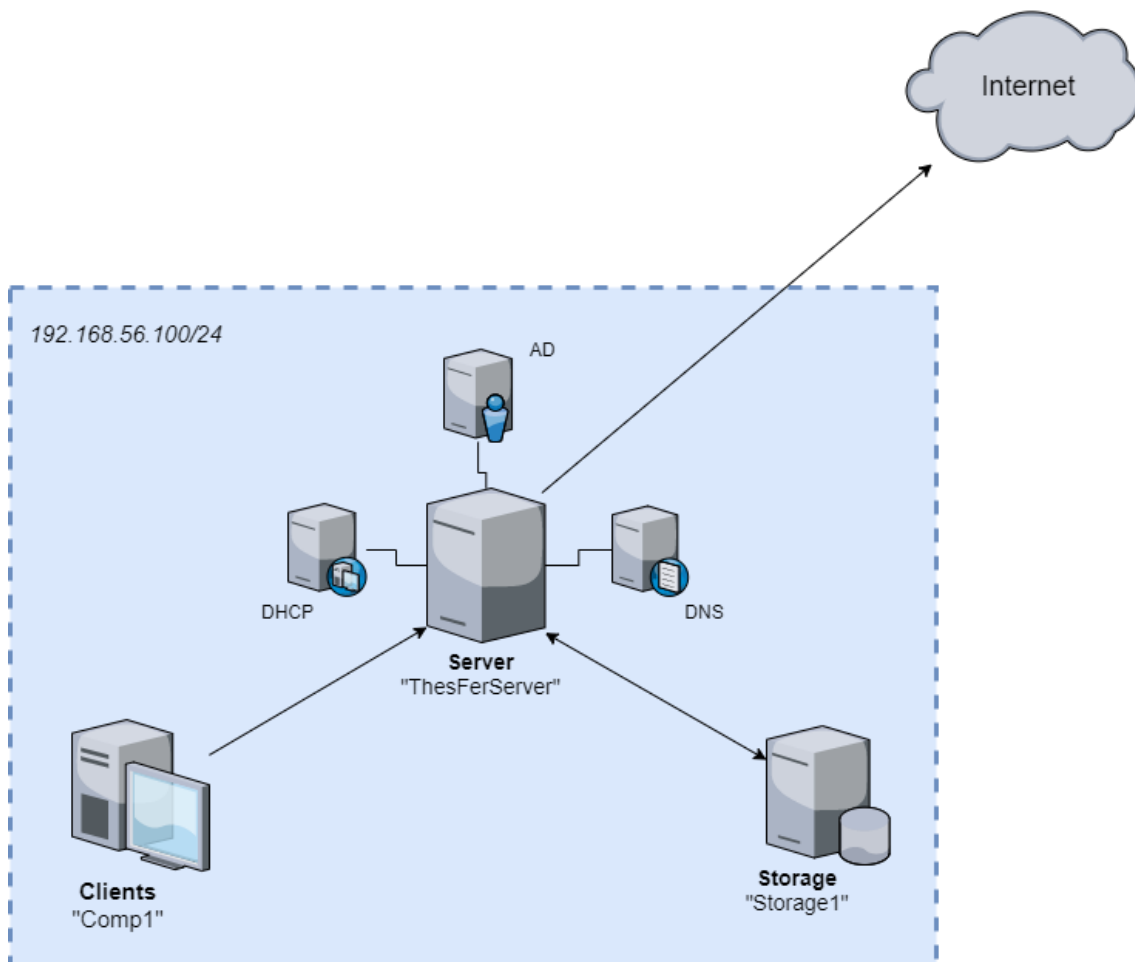


Figure 3. Server architecture.

This simple structure will connect the user's client machine to the server on the same internal network and in the same way to the computer that stores the server's events, also located on the same network. The configuration of these will be explained later in this section.

Once Windows Server has been installed, the different servers are configured. The DNS server was assigned the name "THESFER.local" which will also serve to identify it in the logs. During its configuration, it was verified that the server can issue two sources of logs. The first one provides service information as well as errors at startup or shutdown that is stored within the Event Viewer system. The second one, on the other hand, is recorded in an external file of the administrator's choice where the events of the file transmission are sent after activating the debugging of the server. This option consumes a lot of memory, so it will be kept off until its use is required to test its behaviour in the log collector against log manipulation attacks.

The DHCP server gave some problems at the beginning of the installation due to the configuration of VirtualBox, so the functionality of the virtual machine was used to simulate this same feature, which also allowed access to the IP addresses from the internal server. The range of addresses granted is 192.168.56.100/24 and the server was assigned 192.168.56.102.

Finally, during the configuration of the Active Directory server where the user who will manage the supposed client has been created on the "Comp1" machine (see Figure 3). This user named "user2" has been given permissions to access from a remote desktop and has been assigned to the RW-Shared security group. This group also created on the server, grants read and write permissions to the user, which will allow working with different system alterations, although this is not a realistic feature on a company's server. A more detailed Audit Policy was also established in this server that will allow to obtain more information about different events that occurred in the environment. For this, they based on the recommendations established by Microsoft [32] editing some parameters to cancel the information that is not necessary.

The computer that will act as "client" within the network has been installed with a Windows 10 Enterprise operating system (64 bit)<sup>1</sup>, the evaluation option for virtual machines in its 2002 version of VirtualBox. After connecting it to the DNS server it was assigned the previously created user2 and the IP 192.168.56.103. This user will be used to generate the login and internal network access's events while monitoring this information from the server. Another type of information that can be obtained will be related to the network traffic.

The last element of the network, the collector, will be mounted on a Linux machine with a 64bit Ubuntu 18.04.4 operating system [4]. This device will act as a storage medium for the logs. The IP address assigned in this case is 192.168.56.104. The main function of this device is to record the events sent by ThesFerServer and also to act as a means of visualizing these logs. The reason why a Linux based system was chosen in this case is the ease to configure it and edit the configuration. Its Operating System does not influence the development of the research since the study of manipulation log attacks will be done from Windows environments together with the log shippers.

The summary of the network elements can be seen in the following table:

Table 2. Summary of the elements of the network.

Machine	User	Operative System	IP Address
ThesFerServer	<i>fernando.bauza</i>	Windows Server 2016 Essentials x64	192.168.56.102
Comp1	<i>user2</i>	Windows 10 Enterprise Evaluation x64	192.168.56.103
Collector1	<i>collector</i>	Ubuntu x64 18.04.4	192.168.56.104

Now with all the points of the network installed, the modification of the security settings is complete. As has been said throughout this section, it will allow for more lax protection than would be normal. The access and edition of the different folders have been allowed and the firewall has been configured to allow remote access to the server. This is because we want to simulate a situation where the attacker has already

---

<sup>1</sup> <https://developer.microsoft.com/en-us/windows/downloads/virtual-machines/> [21st of March 2020]

established a connection with the server by some method and has acquired certain permissions to allow him to browse the network at will. The investigation will evaluate the behaviour and response to these attacks by log collectors and not the complexity of defence against them.

## 4.2 Log storage and Log shipping

This project required at least a three-level event transport system, including transfer, storage and display, as outlined in Figure 4

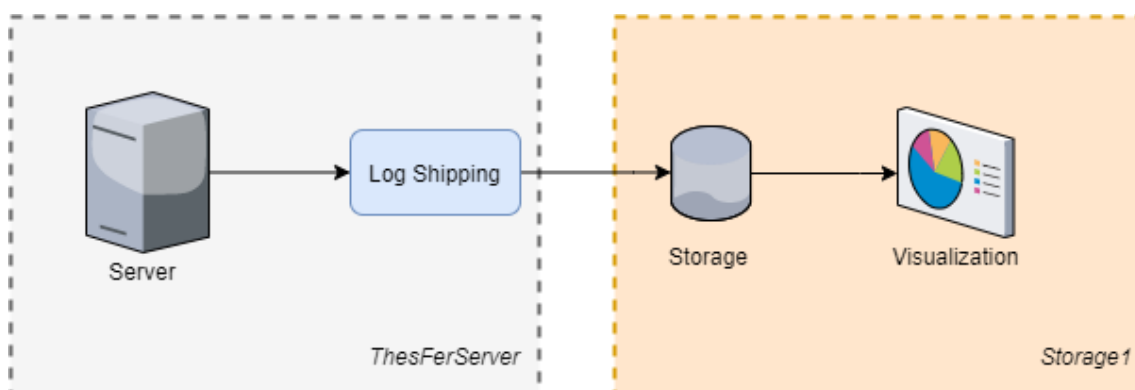


Figure 4. Log transfer esquema.

After the analysis of the different log collectors in the previous section, and their characteristics with which they can affect the effectiveness of log manipulation attacks, we have selected those that will be part of the experimental set-up. Due to hardware limitations, as we are in a test environment built on a single machine, it is not possible to experiment with all the models described above. That is why for this project we have selected from the different options the log shippers belonging to ELK Stack, Winlogbeat and Filebeat to evaluate their behaviour against the manipulation of events in the internal network constituted in ThesFerServer. The selection of Winlogbeat and Filebeat against the other options is due to several factors. Firstly, between them, they cover most of the data sources that we are interested in analyzing such as Windows system events, security and audit events through Winlogbeat, and the events of applications or parts of the server that are hosted in external files to those managed by Event Viewer to which Filebeat has access. On the other hand, being Open Source software and consisting of documentation available on the network, perhaps because of

its greater popularity, it facilitates its installation and configuration on the server. Finally, due to the ease with which these log shippers work with the log storage systems, Elasticsearch, and visualization, Kibana, because they belong to the same family. There exists other structures that also works fine with Elasticsearch but looking at the same software family will avoid possible problems. For these reasons, both programs seem an optimal solution to study in this thesis.

#### 4.2.1 Elasticsearch

Having chosen the log shipper, we proceeded to install the storage for the events in the Storage1 machine, being Elasticsearch the best option to host them. Elasticsearch will act as storage for the events sent by the log collectors Winlogbeats and Filebeats. Some documents [8] prove to be effective for log analysis in small and medium-sized companies, so it is a good choice for this project. Another point to take into account when testing the behaviour of log collectors against log manipulation by potential attackers is the section on transfer to the storage server. In Elasticsearch 7.6 access to the server can be limited to certain interfaces to prevent access from external sources. To connect to the senders installed on the server and have external access without considering the security settings which, as mentioned, will not be taken into account in this project, Elasticsearch has been configured to be accessible and listen from any interface by assigning the address 0.0.0.0 in the *network.host* option. The cluster was assigned the name of Cluster1 and the node1 to make easier its search. The service is accessible through the IP address of the Storage1 machine, 192.168.56.104 and the port 9200 as you can see in Figure 5.

```
root@Collector:~# curl http://192.168.56.104:9200
{
  "name" : "node1",
  "cluster_name" : "Cluster1",
  "cluster_uuid" : "Idr1H7V5SxS2EcYfC6ZmdQ",
  "version" : {
    "number" : "7.6.1",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "aa751e09be0a5072e8570670309b1f12348f023b",
    "build_date" : "2020-02-29T00:15:25.529771Z",
    "build_snapshot" : false,
    "lucene_version" : "8.4.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Figure 5. Elasticsearch

Once the storage service was established, the selected log shippers were installed on the server.

#### 4.2.2 Winlogbeat

Winlogbeat 7.6 allows the extraction of Windows events to send them to the storage system located in our case in Storage1. In its configuration, it was set to ignore events older than 72h, as it is configured by default, to avoid too much noise when analyzing the new events once the tests are done. To select the sources of the logs to be used in the transmission, different parameters were taken into account. Although a more complete configuration would allow a more realistic approach to a small business situation, for a clearer analysis the number of sources has been reduced. On the other hand, if certain events are required that may be interesting to study in case of manipulation. Following indications [33] about events to be evaluated concerning certain scenarios, such as the use of accounts to check unauthorized access and privilege changes in user groups or requests to the DNS server that can identify requests to access malicious sites, Application, Security and System events were added, as well as events coming from the DNS server (Figure 6). Besides, to record some manipulation of events by commands executed in PowerShell, Windows PowerShell and PowerShell/Operational sources were added to keep track of these events.

```
winlogbeat.event_logs:
  - name: Application
    ignore_older: 72h

  - name: System

  - name: Security
    processors:
      - script:
          lang: javascript
          id: security
          file: ${path.home}/module/security/config/winlogbeat-security.js

  - name: Windows PowerShell
    ignore_older: 72h

  - name: Microsoft-Windows-PowerShell/Operational
    ignore_older: 72h

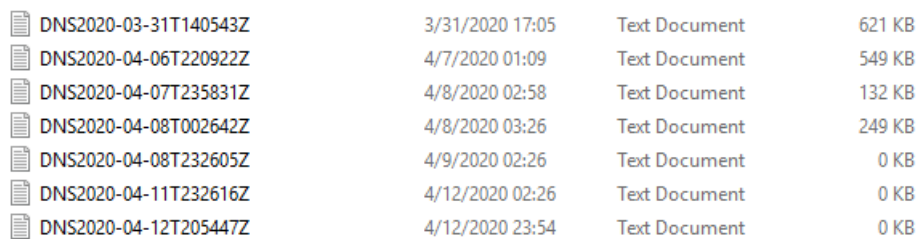
  - name: Microsoft-Windows-DNS-Client/Operational
    ignore_older: 72h
```

Figure 6. Winlogbeat configuration file (winlogbeat.yml).

### 4.2.3 Filebeat

Filebeat 7.6.2 was also installed on the server to transport the logs from files not belonging to the general Windows events to which Winlogbeat does not have access. These files are interesting because unlike many of the usual Windows events they are generally readable without the use of tools like Event Viewer. This can allow an attacker to modify or insert new events if he has the necessary permissions to access and edit the files, thus allowing to trick the user or to remove the trace left behind by his actions on the computer.

To provide Filebeat with a source of events with which to transmit to the Elasticsearch server, the DNS server was configured to perform a debug of the service with which the events caused by the user queries can be seen in greater detail. These events were saved so that files with the date and time were generated as seen in Figure 7, creating different documents according to the days in which the requests were established.



DNS2020-03-31T140543Z	3/31/2020 17:05	Text Document	621 KB
DNS2020-04-06T220922Z	4/7/2020 01:09	Text Document	549 KB
DNS2020-04-07T235831Z	4/8/2020 02:58	Text Document	132 KB
DNS2020-04-08T002642Z	4/8/2020 03:26	Text Document	249 KB
DNS2020-04-08T232605Z	4/9/2020 02:26	Text Document	0 KB
DNS2020-04-11T232616Z	4/12/2020 02:26	Text Document	0 KB
DNS2020-04-12T205447Z	4/12/2020 23:54	Text Document	0 KB

Figure 7. DNS debugging log files.

The location of this folder was assigned in Filebeat as the source of the events, specifying that any document ending in ".log" should be sent to the Elasticsearch server as shown in Figure 8. This provision will also allow testing as will be seen in the next section, due to a weak security configuration that has the purpose of allowing us to test different methodologies without stopping at more sophisticated attacks that may exceed the security of the system.

```

===== Filebeat inputs =====

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

- type: log

# Change to true to enable this input configuration.
enabled: true

# Paths that should be crawled and fetched. Glob based paths.
paths:
  - c:\Users\fernando.bauza\Documents\*.log
  #- c:\programdata\elasticsearch\logs\*

```

Figure 8. Configuration of Filebeat (filebeat.yml).

Finally, for the visualization of all these events that arrive at Elasticsearch's server, a visualization system is required that allows us to analyse the different results obtained. For this purpose, we have used Kibana's system, which belongs to the same Elasticsearch family, so it will facilitate the synchronization with the currently available model. This service was installed in the Storage1 machine allowing access to the events transmitted by Filebeat and Winlogbeat to the server. Through the visualization of the events, it will be possible to first check the state of the event flow in a normal situation, trying different scenarios and actions inside the server in which to study the characteristics before making the manipulations. And secondly, it will be possible to study the efficiency of the attacks carried out at the different points of the transmission and the behaviour of the log collectors in these situations. Kibana was established on port 5601 without establishing any extra security parameters to facilitate its synchronization.

After the final configuration, the architecture of the event flow from the server to the storage centre is as follows:

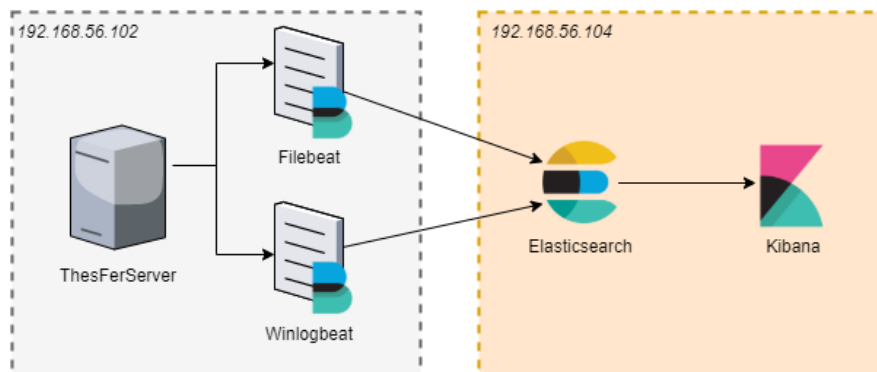


Figure 9. Log flow architecture.



## 5 Experimental Research

With the scenario established in the previous section, at this chapter different log handling systems will be performed in the different transmission states. The objective is to study the behaviour of Winlogbeat and Filebeat in these scenarios and how they affect the normal flow of events that could be expected from a state prior to the attacks. With Winlogbeat we will mainly deal with different ways of altering the logs of the Windows Operating System, such as the login of a user and its consequent Security events or the manipulation of Event Viewer that entails other System events. In the case of Filebeat we will focus on the logs generated by the debugging of the DNS server as shown at the end of the previous chapter.

The events will be transmitted to the Storage1 team where they will be stored in the Elasticsearch platform and can be viewed in Kibana. The effects generated by these attacks can be compared on the Kibana platform. If several records are successfully deleted or their contents manipulated, Kibana will be used to compare the results with the initial state of both the normal flow of events and the actual appearance before their modification. As already mentioned, these models have been made from a low-security system to allow more freedom when testing, it is not a realistic configuration from this point.

To perform the different log manipulation tests, actions will be proposed to edit, generate or destroy the events generated by the different parties. These tasks will be mixed with the normal flow of events trying to deceive the user by omitting information or falsifying it. In a situation where the attacker has accessed the main system with enough privileges to have access to the server, he may try to erase his trace or try to direct the analyst in another direction by using these means. This is because there are different events that can indicate that, although the system is working properly, there could have been a malicious use.

Besides the actions of generating, destroying or modifying the events, the attacks will be oriented towards the three points defined in the section of the log collectors: Attack at the source, attacks in the transit and attacks at the log collector (see Figure 10)

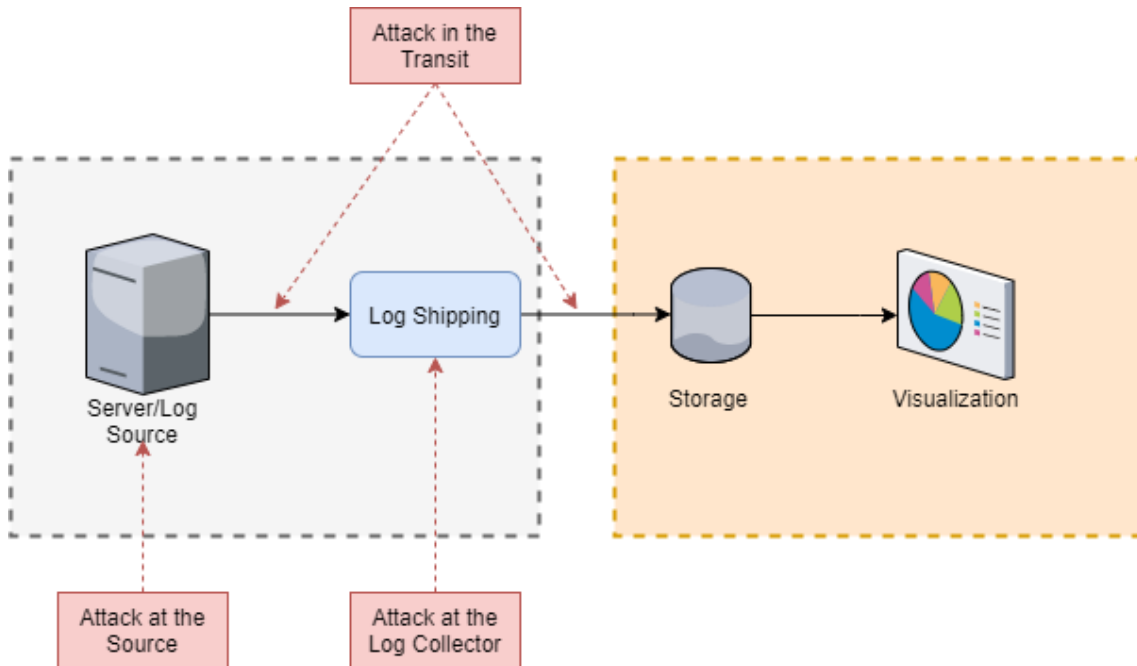


Figure 10. Log attack points.

Regarding the attack directly to the systems of the Beats family (Winlogbeat, Filebeat, etc...) several ideas have been found that can be put into practice in this project. Some examples [11], although applied to Linux systems, deal with some commands to make a delay in sending events to the server that allow the attacker enough time to perform some action before drawing the attention of the filters in Kibana. Although these commands cannot be used in Windows, the same principle is achieved by accessing the configuration manually.

## 5.1 Evaluation on Winlogbeat

On the server, Winlogbeat has been configured to transport System, Security, Application, DNS and PowerShell-related events. All of them can be used to a greater or lesser extent to detect possible attacks on our system [6]. These events are located in the following files:

- C:\Windows\System32\winevt\Logs\Security.evtx
- C:\Windows\System32\winevt\Logs\Application.evtx
- C:\Windows\System32\winevt\Logs\System.evtx
- C:\Windows\System32\winevt\Logs\Windows PowerShell.evtx
- C:\Windows\System32\winevt\Logs\Microsoft-Windows-PowerShell%4Operational.evtx
- C:\Windows\System32\winevt\Logs\Microsoft-Windows-DNS-Client%4Operational.evtx

Saved with the Windows Event Viewer extension ".evtx".

Security events, for example, can indicate information about the use of accounts, such as the event with the ID 4624 that signals successful access to the user account [33]. The occurrence of an event signalling a successful login in a time slot where it should not have existed or using an account that should not be active may indicate unauthorized access to the server's internal network.

In this case, for the analysis of the modification of events in the Source point, we will try to alter this login event from the user2 hosted in the Comp1 machine to the server. First, a normal login has been done, in order to collect the normal state of the event flow that takes place after this action (Figure 11).

>	Apr 15, 2020 @ 16:02:07.774	Logon	4624	user2
>	Apr 15, 2020 @ 16:02:07.772	Kerberos Service Ticket Operations	4769	user2@THESFER.LOCAL
>	Apr 15, 2020 @ 16:02:05.346	Kerberos Service Ticket Operations	4769	user2@THESFER.LOCAL
>	Apr 15, 2020 @ 16:02:05.331	Kerberos Authentication Service	4768	user2

Figure 11. Normal user2 logon events.

In the figure above you can see the user2 login process in which it first requests from the server through the Kerberos service the tickets to certify its authentication and the request to enter the server in the events 4768 and 4769 [5] prior to the event 4624 mentioned above, which signals the successful access to the server. These events would signal the entry of a user to the internal network.

These events have been transmitted to the server at the location of the Windows security events mentioned above. Editing these events is not possible directly, at least not without an external tool, since the EventViewer platform does not allow editing of these events and the events are coded so they are not readable in text format. Therefore, to exercise a manipulation in the source, the removal method will be used to alter the events and prevent them from accessing the display system. To delete the events, we have used the PowerShell terminal and executed the command:

`Clear-EventLog "Security"`

This will eliminate any record placed in the security category, which can be considered excessive or counterproductive from an attacker's point of view but serves to study the behaviour of the log collector.

Note that deleting the logs already transferred by Winlogbeat to the Kibana platform will not prevent them from appearing there, since Winlogbeat does not rewrite the events already sent unlike the display that can be seen in EventViewer. To prevent the transmission of these events, deletion must be done either before the events are transferred or block the communication between the two parties in some way. In this case, it has been chosen to turn off the transmission of the Winlogbeat service and to perform the user login and the deletion of the events during this period.

When the Winlogbeat service is blocked to the Elasticsearch server the events are not recorded so when you log in with user2 and later delete the records of those events, the Kibana dashboard will not be notified of this action. However, other parameters will appear in this display since the actions taken when these commands are executed leave a trace. As can be seen in Figure 12, the events that indicate that the logs have been deleted and that a process has been activated (Winlogbeat) are shown.

>	Apr 15, 2020 @ 23:36:22.882	Apr 15, 2020 @ 23:36:22.882	Security	Process Creation	THESFERSE R\$
>	Apr 15, 2020 @ 23:36:19.291	Apr 15, 2020 @ 23:36:19.291	Security	Log clear	-
>	Apr 15, 2020 @ 23:34:47.345	Apr 15, 2020 @ 23:34:47.345	Security	Logoff	COMP1\$

Figure 12. Visualization status after log cleared and process stopped.

As far as Winlogbeat is concerned, it has not been able to send the user2 login events since it was being blocked when they appeared in EventViewer. Once deleted, although Winlogbeat scans the last events looking for those that have not been sent yet, as in this case the Log cleared and Process Creation events, it is not able to recover those that have already been deleted.

Another way to modify source events is by inserting false events. The insertion of new events in the different folders from which Winlogbeat extracts the logs to transmit to the storage system can mislead the analyst into following false leads or other sources of attack that mask the real problem. Using PowerShell once again we tried to create events that are transported by Winlogbeat.

In this case, it will be written to the System folder, simulating an event with ID 7036 that mentions a process that changes state (in this case winlogbeat), and compared to a similar event already located in Kibana (Figure 13).

```

f log.level          information
f message           The winlogbeat service entered the running state.
f winlog.api        wineventlog
f winlog.channel    System
f winlog.computer_name ThesFerServer.THESFER.local
f winlog.event_data.Binary 770069006E006C006F00670062006500610074002F0034000000
f winlog.event_data.param1 winlogbeat
f winlog.event_data.param2 running
f winlog.event_id   7036

```

Figure 13. Fragment of standard running service information log in Kibana.

The event in the figure above represents the Winlogbeat process going into execution. To try to emulate this event, the following command has been used in PowerShell:

```
Write-EventLog -LogName "System" -Source "Service Control
Manager" -EventID 7036 -EntryType information -Message "The
winlogbeat service entered the running state." -RawData
119,0,105,0,110,0,108,0,111,0,103,0,98,0,101,0,97,0,116,0,47,0,5
2,0,0,0
```

The result (Figure 14) obtained, although certain data tags vary, has managed to reach the Kibana server and there are great similarities with the real event. This type of practice depends on the security with which the execution of commands in the system is maintained and on the filters created in Kibana to evaluate its success. But as far as Winlogbeat is concerned, it has transmitted the event with the indicated parameters having collected it from the System log file.

```
f log.level          information
f winlog.api         wineventlog
f winlog.channel     System
f winlog.computer_name ThesFerServer.THESFER.local
f winlog.event_data.Binary 770069006E006C006F00670062006500610074002F0034
000000
f winlog.event_data.param1 The winlogbeat service entered the running sta
te.
f winlog.event_id    7036
```

Figure 14. Generated System 7036 event.

To affect the events in the Winlogbeat transfer points you can try to block or inject new elements pretending to be the Log Collector to hide the generated record among the existing ones. There are different commands used in Linux to affect the integrity of events in transit [12] to generate packets and send them to the desired destination. In Windows the same model will be reproduced using in this case the PowerShell module

of Elastic.Console<sup>1</sup>. With this extension we can prefabricate events and send them directly to the Elasticsearch indexes. Collecting the index linked to Winlogbeat from the list extracted in Figure 15 has generated an event similar to the 7036 log seen before.

```
PS C:\Users\fernando.bauza\Downloads> es http://192.168.56.104:9200/_cat/indices
green open .kibana_task_manager_1      NcIoNxfSTy48d8d-VBMUw 1 0      2 2    10.8kb  10.8kb
green open ilm-history-1-000001         wGpGmoepQv6kmgve6ZcvQQ 1 0      27 0    41.4kb  41.4kb
green open .apm-agent-configuration    uy0-1nYXTueSX73W_oy9bA 1 0      0 0      283b    283b
yellow open winlogbeat-7.6.1-2020.03.19-000001 A_Aq-AjMRiiBTpdDNlw1aA 1 1    206873 7 225.8mb 225.8mb
green open .kibana_1                    sb1au_FhQW6nef_EYrRwuQ 1 0      62 7    360.9kb 360.9kb
yellow open filebeat-7.6.2-2020.04.12-000001 K3-W7kn0TD21x_52LTASxw 1 1    10422 0      2mb     2mb
PS C:\Users\fernando.bauza\Downloads> _
```

Figure 15. Index list from Elasticsearch.

The timestamp and ID parameters have been changed to avoid conflict with existing events. To also differentiate the new event from those normally collected in Elasticsearch two asterisks have been added at the end of the message. The command (see Appendix 1) uses the HTTP PUSH method to transfer the content to the data store.

The execution proceeded adequately, successfully inserting the event in Elasticsearch as shown in Figure 16. However, this event could not be displayed in Kibana, so the success of the test is in doubt.

```
{ "_index": "winlogbeat-7.6.1-2020.03.19-000001", "_type": "_doc", "_id": "ua-Dj3EB8Sj5s1iIG2fc", "_version": 1, "result": "created",
  "_shards": { "total": 2, "successful": 1, "failed": 0 }, "_seq_no": 208092, "_primary_term": 9 }
PS C:\Users\fernando.bauza\Downloads> S
```

Figure 16. Successful response from Elasticsearch after POST insertion on Winlogbeat index.

Directly affecting the Log Collector to test the third point of attack will somehow affect the accuracy or the ability of the log shipper to do its job. Shutting down the service as was done as part of the source attacks would be a method considered as an attack on Winlogbeat. Another method of manipulation could be applied taking into account the size of the message accepted by Winlogbeat, however this method is conditioned by the maximum size allowed in the system logs.

In this case, in order to attack Winlogbeat directly, it has been decided to manipulate the program's configuration. Changing or deleting the different sources from which Winlogbeat obtains events is the fastest method to prevent certain events from reaching

<sup>1</sup> <https://github.com/elastic/powershell/tree/master/Elastic.Console>

Kibana. To make a more subtle change in the program, the time range where the collector collects the Windows events has been changed. As seen in the Figure 6 most sources have been set to collect events from the previous 72 hours. That parameter has been changed in the Security events to three minutes, allowing certain events to be skipped if they have not been sent previously. To study this effect, we have proceeded as in the manipulation to the source, the Winlogbeat service has been turned off and the configuration has been changed and after 8 minutes the service has been re-established. The results visible in Figure 17 showing how in this case the transmission of the event warning about the deletion of the events has been avoided and no change in the Winlogbeat configuration has been notified. Although the time jump is visible, it can be achieved with greater precision so that it is minimal and does not alter the information of the other events.

>	Apr 17, 2020 @ 01:22:02.091	Apr 17, 2020 @ 01:22:02.091	Security	Special Logon	THESFERSER R\$
>	Apr 17, 2020 @ 01:22:02.091	Apr 17, 2020 @ 01:22:02.091	Security	Logon	THESFERSER R\$
>	Apr 17, 2020 @ 01:16:02.019	Apr 17, 2020 @ 01:16:02.019	Security	Logoff	THESFERSER R\$

Figure 17. Time jump after altering Winlogbeat configuration.

## 5.2 Evaluation on Filebeat

Filebeat allows us to collect the events located in different folders in our system to which Winlogbeat does not have access because they are not part of the Windows logs. In this set-up we will work with the events generated by the debugging of the DNS server that saves its events in the assigned address, C:\Users\fernando.bauza\Documents, with ".log" format. The attacks made on this configuration could also be used in the case of an application that saves its events in its internal folder that would also be connected to Filebeat.

As in the previous case, several tests have been carried out on the different points of the flow covered by the events in which Filebeat may be involved. The objective of the tests is to generate, destroy or modify the events in such a way that they affect in some way the normal response of Filebeat and to be able to analyze the results in Kibana.



In the source the events are saved in ".log" files that are created according to the day and hour when the debugging is activated or if the assigned space in the file runs out. The name with which all the files are generated starts with DNS followed by the date and time and ends with ".log", for example: DNS2020-04-15T002141Z.log. Although Filebeat has been configured to send the logged files for easy testing, a pattern can also be assigned to find the correct events. These logs are sent to Elasticsearch and then to Kibana where they appear as seen in Figure 18.

>	Apr 18, 2020 @ 01:04:27.198	c:\Users\fernando.bauza\Documents\DNS2020-04-15T002141Z.log	4/18/2020 1:04:13 AM 07AC PACKET 0000255BBFDB4F0 UDP Rcv 192.168.56.2 3e84 Q [0001 D NOERROR] A (4)play(6)google(3)com(0)
>	Apr 18, 2020 @ 01:04:27.198	c:\Users\fernando.bauza\Documents\DNS2020-04-15T002141Z.log	4/18/2020 1:04:13 AM 07AC PACKET 0000255BC0469D0 UDP Snd 193.40.56.245 92d8 Q [0001 D NOERROR] A (4)play(6)google(3)com(0)
>	Apr 18, 2020 @ 01:04:27.198	c:\Users\fernando.bauza\Documents\DNS2020-04-15T002141Z.log	4/18/2020 1:04:13 AM 07AC PACKET 0000255BB649140 UDP Rcv 193.40.56.245 92d8 R Q [0001 DR NOERROR] A (4)play(6)google(3)com(0)

Figure 18. Filebeat normal log flow.

The file being manipulated by the server cannot be edited or deleted, so we cannot alter the events directly there. However, it is possible to create a new file in the events folder that matches the Filebeat search pattern [11]. In our case it would be any element ending in ".log" but to make it more difficult to differentiate it has been assigned a name with the pattern similar to the previous events: DNS2020-04-18T231538Z.log. Similar events have been inserted in it, but with another DNS address and changing the sender to the IP address 192.168.56.103 that belongs to user2.

As it is possible to see in the first element of Figure 19, the new event mimics the previous ones generating false information that depending on the filters set by the analyst can lead to deception. Filebeat, by not recognizing the new file as a foreign source to those automatically generated by the server, has transferred the information as easily as it did with the correct events. This type of attack can also be used to massively insert new events into the server to hide or block those that might help uncover the attacker. The success of these measures depends once again on the ability of the intruder to write into the various directories in order to create the fake files that will be transported to Elasticsearch.

```

> Apr 18, 2020 02:32:24.194 c:\Users\fernando.bauza\Documents\DNS2020-04-18T231538Z.log 4/18/2020 2:24:34 AM 07AC PACKET 00000255BC322070 UDP Snd 192.168.56.183 22bc R Q [0001 DR NOERROR] A (1)c(11)s-microsoft(3)com(0)

> Apr 18, 2020 02:31:55.240 c:\Users\fernando.bauza\Documents\DNS2020-04-15T002141Z.log 4/18/2020 2:31:09 AM 07AC PACKET 00000255BCA499A0 UDP Rcv 192.168.56.184 5533 Q [0001 D NOERROR] AAAA (7)appuals(3)com(0)

> Apr 18, 2020 02:31:55.240 c:\Users\fernando.bauza\Documents\DNS2020-04-15T002141Z.log 4/18/2020 2:31:09 AM 07AC PACKET 00000255BC6844B0 UDP Snd 193.40.56.245 a849 Q [0001 D NOERROR] AAAA (7)appuals(3)com(0)

```

Figure 19. Results after log insertion at Filebeat's events source.

The event transfer performed by Filebeat is very similar to the one seen before in Winlogbeat. For this reason, it is unlikely that the same method used previously will work. A similar test will be performed anyway, taking as a body of the message an event extracted from those already sent by Filebeat with the intention of sending it to the Elasticsearch index pretending to be the Log Collector. To do this, the index corresponding to Filebeat has been used with the command used previously (Figure 15) to formulate a command in PowerShell using the Elastic.Console Module to send events to the storage system. The command, which can be seen in Annex 2, sends the event directly to the Elasticsearch page using an unused identifier to avoid collisions with other events.

As can be seen in Figure 20, the shipment has once again been successful, but still no trace appears on the Kibana platform. It is possible to find the event however in Elasticsearch by looking at the index set in the insert. This may be due to a formatting problem where Kibana is not able to differentiate the fields sent to Elasticsearch or the packets are not arriving in JSON format so that they can be read correctly.

```

{"_index": "filebeat-7.6.2-2020.04.12-000001", "_type": "_doc", "_id": "hSOMjXEBRysDwcgLESf-", "_version": 1, "result": "created", "_shards": {"total": 2, "successful": 1, "failed": 0}, "_seq_no": 14206, "_primary_term": 4}
PS C:\Users\Fernando.bauza\Downloads>

```

Figure 20. Successful response from Elasticsearch after POST insertion on Filebeat index.

At the point of attack of the Log Collector, different attacks were tested to check the effect obtained in Filebeat. In this case, it is possible to set events with a larger size in the message field, since we are not limited by the Windows event format. As you can see in the Table 1, the maximum size by default is around 10KiB and 20MiB so we created an event big enough to exceed any of the two measures (26.166KB) in a new



### 5.3 Discussion and Validation

During this research, different tests have been carried out at the points of the event flow through the Log Collector, the source, the transmission and the log shipper. The result of these tests has been verified by displaying the events in their initial state (the insertion) and their final state on the Kibana display medium, where a supposed analyst would study the logs. To assess the changes in the events when they reach their final destination, they have been compared with an initial situation (see Figure 11 and Figure 18) in which no parameters have been altered.

The aim of the situations to which the log shipper has been subjected is to evaluate its behaviour in relation to log manipulation attacks in this Windows environment. The manipulation of the environment has been done without considering the different security elements that protect the system so as not to limit the capacity of these tests. This could consider a situation in which the attacker has acquired high privileges that allow him to access and edit multiple elements of the server without too many obstacles.

It is therefore known that most of the examples established in these experiments are not easily reachable by an external attacker if the security configuration, as well as the different cyphers, were properly configured.

In the attacks to the source of the events, it has been possible to appreciate how both Winlogbeat and Filebeat have transmitted the inserted events as if they were legitimate. Both Log Collectors are dependent on the security of their sources and their configuration specifications to distinguish them when sending legitimate records to the display centre. In the case of Winlogbeat, it is possible to distinguish certain changes in the fields of the inserted log that differ from the generic event. It is possible that this is due to the command used and that with a higher quality program you can solve this problem. Deleted events, while preventable once sent to storage, are not detected if they are destroyed before being collected. Filebeat on the other hand, although in this example it was not possible to edit directly the source file of the events, it depends on the integrity of the folder selected as path and the capacity to filter the files that should be sent to the storage system. By simulating a file with a similarly named pattern, it was possible to send an event with manipulated data to Kibana without being able to distinguish the manipulation beyond the name of the source file.

The results are not so clear in the event transmission point attacks. While it is true that in both cases the transfer of events to Elasticsearch is achieved, they have not been able to access the Kibana display platform. This may be due to the need for a more precise configuration of the JSON packages or the impossibility of entering events directly into the Winlogbeat and Filebeat indexes by this method.

Attacks on the Log Collector itself and its configuration have been more successful in Filebeat than in Winlogbeat. This is mainly due to the restrictions set by the general Windows system events and their format. The configuration manipulation has made it possible to check the limitations of the program when retrieving logs if they exceed the time limit set in the configuration. Filebeat on the other hand, if it has been possible to check the effect of an event with a message greater than the maximum capacity of the log shipper. Once this capacity is exceeded, and without taking into account the limitations of Kibana, Filebeat truncates the event to send it to the database. With this action, there is the possibility of omitting important information that will not be detected in the analysis. On the other hand, there are different fields of its configuration that allow altering the time of data collection of the files in the source. This time allows a strip of action to the attacker that can be used to manipulate the equipment before activating any alarm.

The situations to which both Log Collectors have been subjected have provided multiple data on their behaviour. Although the Windows environment is restrictive with respect to their events and the format they acquire, the results achieved are similar to those described in [12], although other methods have been used, the purpose of the attacks has achieved a comparable objective in the cases of attacks on the source and the log collector to those mentioned. The modifications to the Filebeat configuration described in [11] have also provided results similar to those described.

## 6 Prevention and Detection

Although the aim of this thesis is not to provide complex attacks against log collectors, it can point to certain measures to be considered to protect the service offered by this system and ensure the integrity of the events. Prevention can be applied by making good use of the tools used and their configuration, as well as the security systems already provided by the operating system, or by using tools that add new security functions to fill the gaps left by the above-mentioned ones.

In the case of attacks at the source, consideration must be given to protecting the various locations where events are recorded on the system. Disabling access to and modification of files for most users or roles can prevent many of the attacks made in this environment. The same applies to commands executed in PowerShell. The document [34] suggests some techniques to mitigate their effect on processes.

- Script whitelisting: Can prevent the use of malicious scripts.
- Script execution policy: Designating a secure policy allowing the execution of commands only from reliable and certified sources.
- Powershell version: As described in the above-mentioned paper, PowerShell version 5.0 presents a greater capacity to register the events generated by PowerShell.
- Role-Based application whitelisting: The use of PowerShell and command execution should be restricted to a limited number of privileged users.
- Loggin and Analysis: In case the security measures are exceeded, a detailed analysis of the events and actions performed by PowerShell, such as some of those indicated in the execution of attacks to the source, can alert the user of the malicious actions.

Most of these measures can be applied in the same way to the protection of event files and their directories. Access to only certain roles to these files and a detailed log of the

events generated in these folders would be recommended for further protection against these attacks.

In the case of transmission attacks, it is safest to establish secure communication between both parties. In the experiments carried out, the TLS/SSL protection of Winlogbeat and Filebeat has been deactivated concerning Elasticsearch. An encrypted transmission provides greater protection for the integrity of the messages. The possibility to activate the user authentication system included in the ELK Stack application settings should also be considered. By establishing these users and assigning roles, you can prevent external entities from injecting events into the data flow by generating false logs with which to deceive analysts. Another feature of the configuration used that should be changed for a more secure procedure is to prohibit the client nodes from accessing Elasticsearch and Kibana's HTTP service. This can be done by adding Logstash, that can be used to parse the events, as an intermediary between the Beats tools and Elasticsearch. Also it is recommendable to limit the host network to the ones used to send the data (instead of 0.0.0.0 that creates an open door for every IP). With these measures, some of the possible attacks on the transmission of events during their data flow can be avoided to a greater extent.

Similar measures to those mentioned for source attacks can be used for actions against log collectors. Preventing unauthorized users from accessing the configuration of applications can prevent dangerous attacks against the system. On the other hand, proper monitoring of the events generated in the folders and files related to these elements and a filtering system at the display point can achieve greater protection against attacks that want to supplant legitimate events or take advantage of the operation of log collectors.

With regard to the different signs indicating the presence of the manipulation of events in the cases studied, a pattern could be seen in all of them.

In the case of the events at the source, the presence of the log is clear, indicating that the system events have been deleted or that a new process has been started, either Winlogbeat or Filebeat. This signal can alert to this intentional manipulation of the source of events. The same applies to attacks on the Winlogbeat configuration where, although the event indicating the deletion of the source file was not collected, the event

of the new Winlogbeat process appears. In the case of the Filebeat event source, the falsification of data can be detected by taking into account the appearance of two different log file names interspersed in the same period of time. It is also remarkable in the case of the falsification of log files with the same name but with a different location, that if we filter by the address of the sources the new manipulated source is highlighted. The test that performed a truncation of the event from exceeding the maximum capacity of the message, although it could not be seen in Kibana because it also exceeded its configuration as it can be seen in Figure 21 how a flag is activated indicating its abnormal state.

All this data can be used to detect similar attacks by an analyst. The detection of these manipulations can prevent a possible attack that is being carried out or that has been carried out recently, turning the attempt to cover its track of the attacker in an alert that warns of the security violation.

Some tools can enhance event integrity protection and prevention systems. Implementing a log signature scheme [13] could ensure the integrity of the entire event in any part of the data flow. This kind of measures has already been included in log collectors such as Rsyslog to exercise this security function on the events handled by this application. Other protection methods that have been studied are based on the strengths of the blockchain to ensure the integrity of the events [14]. Some research [15] proposes systems based on this technology, seeking the immutability of the protected events. With the help of the blockchain, the security of the forensic evidence obtained in the events is improved [16].

Finally, structures have also been investigated that facilitate the verification of the events handled in an efficient manner [17]. This report proposes a data structure called concurrent authenticated tree that proves the verification of hash-based events in a more efficient way.

Protection techniques such as the event signature mentioned above can be very effective in protecting against attacks on the source where the attacker tries to inject new events or introduce a non-legitimate source. Also, both signed events and blockchain systems should work particularly well in transmission attacks since any changes, if any, will be detected by the system and recorded. Attacks on the log collector can be somewhat



more complex to deal with, most of which can be caused by a bad configuration or by elements that are beyond the capacity of the tool. In these situations, it is advisable to have a greater number of indicators and filters that allow the detection of variations and flags that alert of these eventualities.

## 7 Conclusion

In the internal networks at companies whose events coming from multiple sources and devices are centralized to monitor the events, the protection of the logs becomes important for them to be able to perform their job. The log collectors handle these events generated by the different sources distributed throughout the internal network and are the target of attacks that attempt to manipulate the information provided by the logs to hide the attacker's true intentions and deceive the analyst. The study of the behaviour of log collectors in the context of this type of attack on Microsoft networks provides new information to help detect these situations that may be executed at different vulnerable points in the data flow.

In this thesis, different ways of attacking the event transport have been sought, having identified the points of source, transit, and the log collector itself as the sections vulnerable to these attacks. At these points, events have been generated and/or eliminated to alter the information displayed in Kibana and hide a possible attack. Following the results of the experiments, we consider the hypothesis that these attacks carried out on the vulnerable points of the process performed by the log collectors allow manipulation of the information tested at most of the points.

The results obtained at the source of the events, allowing the insertion and deletion of some logs, show how these changes arrive at Kibana managing to omit important information or showing false data with a format similar to the real ones. The same happens in the attacks to the log collector, where it has been possible to see how a direct manipulation of the configuration can allow the omission of events with even more precision. The results are more evident in Filebeat since they allow more freedom in the source of the events and where it has been possible to reach the maximum size of the message, thus sending a truncated event that reduced the information that the log should expose. In the transport however, the results have not been so clear, and cannot be considered valid concerning the parameters established for the validation of the results.

Different traces have also been identified that may lead to the detection of these manipulation attempts. Attacks at the source leave traces indicating a possible manipulation such as events indicating that logs have been deleted or that a new process has been started (this being Winlogbeat or Filebeat). Events collected by Filebeat from a different location leave a trace of that address and file name. The truncated event, on the other hand, is sent together with a flag that warns of this situation, so it could also be filtered, and this manipulation detected.

The evaluation of the limits of Winlogbeat's capacity for event transport has been limited by the event format of the Windows OS. In Filebeat if it has been possible to do so, checking how, although reduced, it continues to send the event to Elasticsearch, warning of its status.

The method used in this investigation by identifying the vulnerable parts in the transport of the events has allowed the classification of the attacks and the respective reactions in a more localised way, analysing processes similar to the situation and the work carried out by the log collector at each moment. Although the development in a small company scenario has allowed a clearer vision of the attacks for their identification, it also makes it difficult to measure them in busier environments with more powerful attacks. On the other hand, as it has been said throughout this project, the results have been collected in an environment with a minimum security configuration, so it is not intended to give validity to the attacks carried out but to how the log collectors have reacted to them.

This thesis provides some more information to this gap in the literature, where analysis reports on log collectors in response to this type of attack are very scarce. The results obtained may provide more information for event analysts and threat hunting services to find attack patterns that may be similar to those identified in this thesis. The outcomes reached may also be useful for the development of protection methods that consider the vulnerable sectors that have been identified in the course of the events to the storage system.

In **future researches**, this data can be used to expand to other types of log collectors using a similar methodology. The strength of the different buffer systems that each software handles can also be tested to try to exceed their capacity, thus testing possible new ways to affect event monitoring after causing a buffer overflow that could stop

transmission or omit some of the logs. Another interesting field to deal with would be the study of its behaviour in log poisoning attacks, where executable code is injected into the events so that it is activated remotely or automatically. This field has also been studied in more detailed way on Linux networks leaving Microsoft networks without much information about it.

## References

- [1] “Common Attack Pattern Enumeration and Classification (CAPEC),” MITRE, 30 September 2019. [Online]. Available: <https://capec.mitre.org/data/definitions/93.html>. [Accessed 04 April 22].
- [2] “Security and More,” 19 May 2018. [Online]. Available: <https://liberty-shell.com/sec/2018/05/19/poisoning/>. [Accessed 22 April 2020].
- [3] K. Kent and M. Souppaya, Guide to computer security log management., Gaithersburg: NIST special publication 92, 2006.
- [4] C. Lobo, “Security Log Management,” *Network Security*, vol. 2003, no. 11, pp. 6-9, 2003.
- [5] Information Assurance Directorate, “Spotting the Adversary with Windows Event Log Monitoring,” 2013.
- [6] R. Anthony, Detecting Security Incidents Using Windows Workstation Event Logs, SANS Institute, 2012.
- [7] J. Hamilton, B. Schofield, M. Gonzalez Berges and J.-C. Tournier, “SCADA Statistics Monitoring Using the Elastic Stack (Elasticsearch, Logstash, Kibana),” in *TUPHA034*, Barcelona, 2017.
- [8] S. J. Son and Y. Kwon, “Performance of ELK Stack and Commercial System,” in *IEEE 13th Malaysia International Conference on Communications (MICC)*, Malaysia, 2017.
- [9] A. Messina, I. Fontana and G. Giacalone, “Log monitoring and analysis with rsyslog and Splunk,” Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), Itlay, 2015.
- [10] Palantir, “Windows Event Forwarding for Network Defense,” Medium, 11 September 2017. [Online]. Available: <https://medium.com/palantir/windows-event-forwarding-for-network-defense-cb208d5ff86f>. [Accessed 22 April 2020].
- [11] B. McIver, “Abusing Elastic’s Beats to Avoid Detection and Manipulate Logging,” RIT Computing Security Blog, 7 December 2017. [Online]. Available: <https://ritsec.wordpress.com/2017/12/07/abusing-elastics-beats-to-avoid-detection-and-manipulate-logging/>. [Accessed 22 April 2020].
- [12] A. Chuvakin, K. Schmidt and C. & Phillips, Logging and log management: the authoritative guide to understanding the concepts surrounding logging and log management., Newnes, 2012.
- [13] A. Buldas, A. Truu, R. Laanoja and R. Gerhards, “Efficient Record-Level Keyless Signatures for Audit Logs,” in *Nordic Conference on Secure IT Systems*, Springer, Cham, 2014.
- [14] L. Aniello, R. Baldoni, E. Gaetani, F. Lombardi, A. Margheri and V. Sassone, “A Prototype Evaluation of a Tamper-resistant High Performance,” in *13th European Dependable Computing Conference*, 2017.

- [15] W. Pourmajidi and A. Miranskyy, “Logchain: Blockchain-assisted Log Storage,” in *2018 IEEE 11th International Conference on Cloud Computing*, San Francisco, CA, USA, 2018.
- [16] P. T. Duy, H. Do Hoang, D. Thi Thu Hien, N. Ba Khanh and V.-H. Pham, “SDNLog-Foren: Ensuring the Integrity and Tamper Resistance of Log Files for SDN Forensics using Blockchain,” in *6th NAFOSTED Conference on Information and Computer Science (NICS)*, Hanoi, Vietnam, 2019.
- [17] F. Ning, Y. Wen, G. Shi and D. Meng, “Efficient tamper-evident logging of distributed systems via concurrent authenticated tree,” in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, San Diego, CA, USA, 2017.
- [18] “Sandfly Security,” Sandfly Security, 1 August 2019. [Online]. Available: <https://www.sandflysecurity.com/blog/using-linux-utmpdump-for-forensics-and-detecting-log-file-tampering/>. [Accessed 22 April 2020].
- [19] Graylog, “CENTRALIZED LOGGING – KNOWING WHEN LESS IS MORE,” Graylog, 14 January 2019. [Online]. Available: <https://www.graylog.org/post/centralized-logging-knowing-when-less-is-more>. [Accessed 22 April 2020].
- [20] Rsyslog, “The rocket-fast Syslog Server,” 2020. [Online]. Available: <https://www.rsyslog.com/>. [Accessed 22 April 2020].
- [21] R. Gerhards, “Reliable Forwarding of syslog Messages with Rsyslog,” 27 June 2008. [Online]. Available: [https://www.rsyslog.com/doc/v8-stable/tutorials/reliable\\_forwarding.html](https://www.rsyslog.com/doc/v8-stable/tutorials/reliable_forwarding.html). [Accessed 22 April 2020].
- [22] R. Gerhards, “Legacy Global Configuration Statements,” [Online]. Available: <https://www.rsyslog.com/doc/v8-stable/configuration/global/index.html>. [Accessed 22 April 2020].
- [23] R. Gerhards, “General Queue Parameters,” [Online]. Available: [https://www.rsyslog.com/doc/master/rainerscript/queue\\_parameters.html](https://www.rsyslog.com/doc/master/rainerscript/queue_parameters.html). [Accessed 22 April 2020].
- [24] Syslog-ng, “syslog-ng Open Source Edition 3.19 - Administration Guide,” [Online]. Available: <https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition/3.19/administration-guide/48#TOPIC-1094644>. [Accessed 22 April 2020].
- [25] Microsoft, «Use Windows Event Forwarding to help with intrusion detection,» 28 02 2019. [En línea]. Available: <https://docs.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection>. [Último acceso: 22 April 2020].
- [26] Elastic, “Winlogbeat Reference,” [Online]. Available: <https://www.elastic.co/guide/en/beats/winlogbeat/current/index.html>. [Accessed 22 April 2020].
- [27] Elastic, “winlogbeat.reference.yml,” [Online]. Available: <https://www.elastic.co/guide/en/beats/winlogbeat/master/winlogbeat-reference.yml.html>. [Accessed 22 April 2020].
- [28] Elastic, “Filebeat Reference,” [Online]. Available: <https://www.elastic.co/guide/en/beats/filebeat/master/index.html>. [Accessed 22 April 2020].
- [29] Securing Splunk Enterprise, “About securing data from forwarders,” 07 August

2019. [Online]. Available:  
<https://docs.splunk.com/Documentation/Splunk/8.0.2/Security/Aboutsecuringdatafromforwarders>. [Accessed 22 April 2020].
- [30] Microsoft, “Appendix L: Events to Monitor,” 30 July 2018. [Online]. Available:  
<https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/appendix-l-events-to-monitor>. [Accessed 22 April 2020].
- [31] RSA Information Design and Development , “Log Collection: The Basics,” 04 May 2017. [Online]. Available: <https://community.rsa.com/docs/DOC-42977>. [Accessed 22 April 2020].
- [32] Microsoft, “Audit Policy Recommendations,” 31 May 2015. [Online]. Available:  
<https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations>. [Accessed 22 April 2020].
- [33] nsacyber, “Windows Event Monitoring Guidance,” Github, 2019. [Online]. Available: <https://github.com/nsacyber/Event-Forwarding-Guidance/tree/master/Events>. [Accessed 22 April 2020].
- [34] ACSC, «Australian Cyber Security Centre,» March 2019. [Online]. Available: [https://www.cyber.gov.au/sites/default/files/2019-03/Securing\\_PowerShell.pdf](https://www.cyber.gov.au/sites/default/files/2019-03/Securing_PowerShell.pdf). [Accessed 16 May 2020].

## Appendix 1 – PowerShell command for log injection on Winlogbeat Transit

```
es -Method POST "http://192.168.56.104:5601/winlogbeat-7.6.1-2020.03.19-000001/_doc/ua-DJ3EB8Si5sliIG2fc" -Body @'
{
  "source": {
    "@timestamp": "2020-04-16T18:17:13.140Z",
    "winlog": {
      "api": "wineventlog",
      "process": {
        "pid": 552,
        "thread": {
          "id": 5621
        }
      }
    },
    "record_id": 9195,
    "task": "",
    "computer_name": "ThesFerServer.THESFER.local",
    "keywords": [
      "Classic"
    ],
    "provider_guid": "{555908d1-a6d7-4695-8e1e-26931d2012f4}",
    "event_data": {
      "param2": "running",
      "Binary": "770069006E006C006F00670062006500610074002F0034000000",
      "param1": "winlogbeat"
    },
    "channel": "System",
    "event_id": 7036,
    "provider_name": "Service Control Manager"
  },
  "event": {
    "created": "2020-03-29T18:17:14.568Z",
    "kind": "event",
    "code": 7036,
    "provider": "Service Control Manager"
  },
  "host": {
    "architecture": "x86_64",
    "os": {
      "version": "10.0",
      "family": "windows",
      "name": "Windows Server 2016 Essentials",
      "kernel": "10.0.14393.2248 (rs1_release.180427-1804)",
      "build": "14393.2248",
      "platform": "windows"
    },
    "id": "76fca97e-d16f-4915-949b-87c18b61352d",
    "name": "ThesFerServer.THESFER.local",
    "hostname": "ThesFerServer"
  }
}
```



```

    },
    "ecs": {
      "version": "1.4.0"
    },
    "agent": {
      "type": "winlogbeat",
      "ephemeral_id": "2f82820c-a356-42ee-9297-0db28baa3013",
      "hostname": "ThesFerServer",
      "id": "acbc855e-107c-4953-b40f-8b416ee7672c",
      "version": "7.6.1"
    },
    "log": {
      "level": "information"
    },
    "message": "The winlogbeat service entered the running state.**"
  },
  "fields": {
    "@timestamp": [
      "2020-04-16T18:17:13.140Z"
    ],
    "event.created": [
      "2020-04-16T18:17:14.568Z"
    ]
  },
  "sort": [
    1585505833140
  ]
}
'@

```

## Appendix 2 – PowerShell command for log injection on Filebeat Transit

```
es -Method POST "http://192.168.56.104:9200/filebeat-7.6.2-2020.04.12-000001/_doc/hSOMjXEBRysDWcgLESf_" -Body @'
{
  "source": {
    "@timestamp": "2020-04-18T13:48:08.210Z",
    "log": {
      "offset": 119183,
      "file": {
        "path": "c:\\Users\\fernando.bauza\\Documents\\DNS2020-04-18T130503Z.log"
      }
    },
    "message": "4/18/2020 4:47:35 PM 0BE0 PACKET 000001E2A7F705B0 UDP Rcv 192.168.56.102 1bbe Q [0001 D NOERR OR] A (8)accounts(6)google(3)com(0)",
    "input": {
      "type": "log"
    },
    "ecs": {
      "version": "1.4.0"
    },
    "host": {
      "architecture": "x86_64",
      "os": {
        "platform": "windows",
        "version": "10.0",
        "family": "windows",
        "name": "Windows Server 2016 Essentials",
        "kernel": "10.0.14393.2248 (rs1_release.180427-1804)",
        "build": "14393.2248"
      },
      "id": "76fca97e-d16f-4915-949b-87c18b61352d",
      "name": "ThesFerServer",
      "hostname": "ThesFerServer"
    },
    "agent": {
      "hostname": "ThesFerServer",
      "id": "83f51e85-67f5-4d41-9f77-06e278cdb219",
      "version": "7.6.2",
      "type": "filebeat",
      "ephemeral_id": "a4a474a9-ab33-4fac-95ba-731a00900262"
    }
  },
  "fields": {
    "suricata.eve.timestamp": [
      "2020-04-18T13:48:08.210Z"
    ],
    "@timestamp": [
      "2020-04-18T13:48:08.210Z"
    ]
  }
}
```

```
]
},
"sort": [
  1587217688210
]
}
'@
```