Observer effect: How Intercepting HTTPS traffic forces malware to change their behavior

María José Erquiaga[1,2,3] , Sebastián García[4] and Carlos García Garino[1,3],

[1] ITIC, UNCuyo, Mendoza, Argentina
[2] Facultad de Ciencias Exactas y Naturales, UNCuyo, Mendoza, Argentina
[3] Facultad de Ingeniería, UNCuyo, Mendoza, Argentina
merquiaga@uncu.edu.ar, cgarcia@itu.uncu.edu.ar
[4] CTU University, Czech Republic
sebastian.garcia@agents.fel.cvut.cz

Abstract. During the last couple of years there has been an important surge on the use of HTTPs by malware. The reason for this increase is not completely understood yet, but it is hypothesized that it was forced by organizations only allowing web traffic to the Internet. Using HTTPs makes malware behavior similar to normal connections. Therefore, there has been a growing interest in understanding the usage of HTTPs by malware. This paper describes our research to obtain large quantities of real malware traffic using HTTPs, our use of man-in-the-middle HTTPs interceptor proxies to open and study the content, and our analysis of how the behavior of the malware changes after being intercepted. The research goal is to understand how malware uses HTTPs and the impact of intercepting its traffic. We conclude that the use of a interceptor proxy forces the malware to change its behavior and therefore should be carefully considered before being implemented.

Keywords: Malware, Botnets, HTTPs, Malware Traffic, Network Security, MITM, proxy, Malware Behavior

1 Introduction

In physics, the observer effect alludes to the influence of the observer on the phenomenon under observation. In most areas, this influence is often caused by instruments and modifies the behavior of what is being measured in some important manner. However, the observer effect can also appear in other contexts, such as malware execution and analysis for network security. Unfortunately, the impact of capturing techniques on the behavior of malware in the network was often overlooked. In this work, we study the observer effect in regard to the use of web TLS (Transport Layer Security) interceptor proxies for network malware analysis. The usage of web proxies TLS interceptors allows companies to view the content of its employees encrypted connections. This technology is nowadays widely implemented in large companies and it is supposedly used to protect the employees from infections. However, the inflicted changes upon the behavior of malware, the invasion of

employee's privacy and the implementation costs can be an inconvenience for some companies.

This research focuses on understanding the influence of web proxies TLS interceptors during the execution of malware. To achieve the analysis, we created a new and large TLS dataset of real malware samples extending for more than one year. The malware was run in a special infrastructure for selective interception and with complete access to the Internet. The average execution time of each malware is one week, with some executions lasting two months. The main goal of this work is to find how malware changed its behavior due to the filtering, blocking and interception of its web TLS connections. The most important factor of our research was the verification process, which consisted in executing the malware twice and simultaneously: once using the interceptor proxy and once without it.

The contributions of this work are: (1) The creation of a dataset. The dataset created is a large and modern dataset, it includes more than 80 malware captures and captured carefully with a methodology explained in this article. (2) Malware interception using https or port 443. The malware selected to captured was the one using encrypted communication or port 443 (the port assigned to encrypt the communication). Also, two scenarios were considered; with proxy and without proxy. (3) Publication of the dataset for the research community. The dataset was published in the stratosphere web site and twitter. (4) An analysis of the implications of intercepting the traffic of malware (with and without proxy), described in this paper.

The remainder of this work is organized as follows. Section 2 describes the background concepts, definitions and the previous work. Section 3 describes the dataset creation and features. This section also includes a description of the laboratory infrastructure and the methodology to obtain the dataset. Section 4 shows an analysis of the malware captures and a comparison of the same malware with and without using proxy. Finally, Section 5 presents the conclusions and future work.

2 Background and Previous Work

TLS is the standard security protocol for encrypting information in a network. It establishes an encrypted link using asymmetric and symmetric encryption algorithms. TLS is the widest used encryption protocol because it uses a trust chain to verify that the service is trustable and it doesn't require the client to have any special password. Recently, there seems to be a rise in the use of TLS by malware, causing new difficulties for the analysts and rising several new questions about what the malware is doing inside the encrypted channel, and how its attack strategy changes.

The study of TLS interception in the network and its impact on the users has been analyzed before from the perspective of privacy [1]. The practice of using TLS interceptor proxies is common inside companies, even when employees are unaware. The most important impact of this practice is that users get used to being intercepted and their general security protection measures decrease. The authors concluded that approximately 0.6% of all users inside companies are subject to interception.

The analysis of TLS usage by malware was previously studied by Cisco in an attempt to find their actions without decrypting the traffic [5]. The authors used 18

malware families that usually encrypt their traffic to understand the motivations behind encryption. The research found that 98.4% of the encrypted malware traffic used port 443/TCP for this purpose. However, there is no analysis of the malware using port 443/TCP with other protocols that are *not* TLS. Therefore, they mainly focused on TLS protocol on port 443/TCP.

Furthermore, Carné de Carnavalet et al [2], conduct investigations into the risks introduced by TLS interception tools, by analyzing the risks of using antivirus and parental control tools with a proxy.

Cisco researchers have found that the amount of malware using TLS has grown in the last years. Their analysis reveals that malware uses TLS differently than normal traffic. [6]. The author proposed to augment the common 5-tuple structure with TLS-based features. Among the additional features are the list of offered cipher suites, the selected cipher suite, the sequence of lengths and type codes of TLS records, and the time between TLS records in milliseconds. [7]

The most common analysis of the TLS protocol on malware traffic focus the detecting the malware. Common techniques include, for example, behavioral traffic analysis using a k-NN classification [3]. Another recent approach was to detect the malware by studying its encrypted HTTPs communication [4]

3 HTTPs Malware Dataset

The most important challenge of analyzing malware using HTTPs is the lack of a good public dataset. As part of our work we spent almost one year collecting real, and long term, malware traffic. The dataset created is part of our Nomad Project [9]. Our dataset consists of more than 80 network malware traffic captures. One of the goals of the dataset is to study the behavior of malware and how it changes in time. To obtain this type of data we executed the malware for long terms, up to 3 weeks or even months. The dataset contains malware capture of different types of malware (such as Trojans, Adware, botnets, etc.). For each capture, we generated several files to improve future analyses.

The process of creating the dataset can be described in four phases, (1) design and creation of the laboratory, (2) design of the capture methodology, (3) generation of experiments and output of information. The following subsections describe these phases in detail.

3.1. Malware Laboratory Infrastructure

The malware laboratory infrastructure consisted in a host Linux computer running more than 30 VirtualBox Windows 7 virtual machines. The host computer also ran a separate mitmproxy[1] implementation for each malware, allowing the complete isolation of results. The malware had unrestricted access to the Internet except for a limited bandwidth and an SMTP redirection to an e-mail honeypot. Fig. 1 shows a

[1] https://mitmproxy.org/

basic schema of the intercepting infrastructure. It can be seen how all most common ports for the web connections were redirected through the web TLS interceptor.

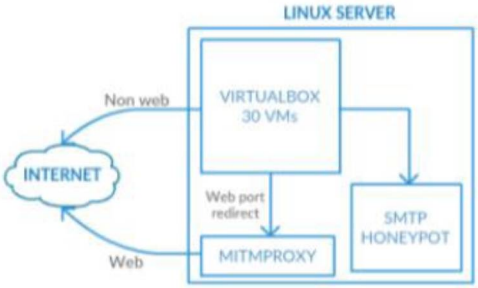

Figure 1. Malware capture laboratory infrastructure

3.2 Capture Methodology

The methodology implemented to capture malware traffic is based on the analysis of known malware that we use to infect machines in our laboratory, described before. The steps to capture malware traffic are: (1) find malware binary (2) copy the binary to the server (3) start the virtual machine and infect it (4) compute the start date and the infection date and monitoring the machine (5) stop and publish the capture. To find the malware binaries, we search in web sites specialized in malware samples such as virus total, hybrid analysis, and SSL blacklist. We monitor each machine using tcpdump, mitmproxy outputs and cacti. Thought those tools it was possible to detect if the machines were infected or not and if the communication was still alive. Once we considered we had enough information, we stopped the capture, generate the output of the dataset and published it in with the corresponding ID in the stratosphere website, blog and twitter.

3.3 Experiments and output files

The dataset consists of 80 different malware captures [8]. The malware captured includes botnet, trojans, backdoors and adware among others. Approximately 90% of these captures used port 443/TCP to communicate. However, only 83% of the captures used the TLS protocol over port 443/TCP. This means that at least 17% of malware used port 443/TCP for their own custom encryption protocol. Regarding the usage of these connections, from the complete set of captures, approximately 30% used the port 443/TCP (both with and without TLS) for their Command and Control channels. This means that most of the TLS connections generated by malware were not directly related to its remote control.

Each malware has assigned a unique ID. This ID allow us to identify each malware capture. If we execute the malware twice, we add a number to the ID. For instance, 260 is the capture ID, the first capture will be 260-1, the second one 260-2, and so on. The URL format to access to the capture information contains this ID. For example: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-260-1.

4 Comparison and Evaluation of the Behavior of Malware with Intercepting TLS proxies.

In this section, we analyze the network behavior of malware captures when they are intercepted by an TLS proxy. We analyzed each malware traffic capture to evaluate the behavior in two scenarios: with proxy interception and without proxy interception. The goal is to detect the behavior differences in the same malware when the execution was made with or without proxy. As we explained in **Section 3.3**, each capture has a unique ID, the purpose of this ID is to identify each capture. For instance, the malware capture ID 169-1 belongs to a malware capture, the ID 169-2 is the same malware executed a second time. We will analyze malware captures with proxy and without proxy. Therefore, ten malware captures were analyzed: 169-1, 169-2, 192-1, 192-2, 219-1 and 219-2.

4.1 Miuref Capture (169-1)

This malware belongs to the Miuref family. It was executed three times, twice with an interceptor proxy (captures 169-1 and 169-2) and once without proxy (capture 169-3).

Miuref Capture with TLS Proxy
In this capture, Miuref first resolves the domains service8.org, 1.web-counter.info, timeservice24.com, 2.web-counter.info, 3.web-counter.info, 4.web-counter.info, and 5.web-counter.info. These are the main Command and Control (CC) channels of the malware. All the domains have one or more IP addresses and the malware starts to connect to them on port 443/TCP. Given that the protocol spoken by the malware on port 443/TCP is not TLS, the MITM proxy generates the follow error:

502 Bad Gateway. SSL handshake error: The client may not trust the proxy's certificate.

This is the first important conflict between the malware and the proxy, since the proxy does not allow the traffic to connect to Internet. Therefore, the C&C servers are not reached properly. After the unsuccessful connections, the malware contacts an IP address that is hardcoded in the binary: **185.118.67.195**. This IP is contacted on port 80/TCP and the protocol spoken is correct HTTP. This backup C&C mechanism worked well and allowed the malware to download a possible binary update.

The second most important conflict of the malware and the MITM proxy is that the malware keeps trying to connect to the 443/TCP ports in the C&C servers, generating a large amount of very suspicious traffic.

As the malware is related with ClickFraud and advertising, it also connects to twitter.com, youtube.com, google.com, facebook.com, bing.com, etc. These requests were done because certain webpages had links to them and not because the malware was abusing their services. We were able to verify this claim because it was possible to observe the HTTPs traffic and distinguish the real requests.

Then, the malware resolves the domain bam.nr-data.net, obtaining the following IPs 50.31.164.175, 162.247.242.18, 50.31.164.173, 50.31.164.166, 50.31.164.174, 162.247.242.19. It uses port 80 to contact the IP 50.31.164.175 and establishes a CC channel. For example, an unsuccessful connection to port 443 that were not SSL is the one to the IP 138.201.125.95.

Miuref Capture Without Proxy (169-2)

This is a capture of the Miuref malware but without using any MITM proxy. In this case, the malware tries to connect to its C&C servers using the port 443/TCP with its custom encryption protocols and it is successful. After contacting the C&C servers the malware also contacts its hardcoded IP address **185.118.67.195** and downloads a binary update. The operation of the malware remains similar to the capture with the proxy and it connects to several ClickFrauds and advertising sites (platform.twitter.com, connect.facebook.net, google.com and bing.com).

In this capture the behavior is similar to the previous capture (resolves the same domains, contacts the same IPs). However, the main difference is that the TLS connection is established. The differences in domains accessed by the two captures is also significant, although it is difficult to know exactly why the malware is connecting to hundreds of sites. From the 465 totals, unique domains accessed by the capture with MITM proxy and the capture without MITM proxy, 86% was requested by the malware without the MITM proxy. From the 206 unique domains requested in the capture without MITM proxy, 30% were unique to this capture. From all the traffic in the capture with the MITM proxy, 18% was sent to port 443/TCP, while only 8% of the traffic in the capture without the MITM proxy was sent to the same port. This difference is due to the use of an interceptor HTTPs proxy.

4.2 Remote Admin.Ammyy Capture (ID 192)

This malware is a remote admin tool called Ammyy. Ammyy is a possible legitimate application, but it is usually abused for its remote administration capabilities. This malware was executed twice; with proxy, and without proxy.

Remote Admin.Ammyy Capture 192-2 with proxy

This malware first resolves the domain *rl.ammyy.com* and contacts the IP **176.56.184.37** in port 80/tcp. Then, it contacts four different IPs in the ports 443/TCP, 80/TCP and 8080/TCP in sequence. The IP address are **88.198.6.56**, **88.198.6.55**, **95.211.242.83** and **95.211.191.142**. Those IPs could be hardcoded in the binary file, or they may be delivered in the communication C&C channel with the server from the IP 176.56.184.37. The malware does not connect successfully to the IPs 88.198.6.56, 88.198.6.55, 95.211.242.83 and 95.211.191.142. This happens because the MITM

proxy cannot recognize the protocols used in those ports. It means, that the malware is using the protocol in an unconventional manner.

Remote Admin.Ammyy Capture 192-1 and 192-3 without proxy
The malware resolves the domain rl.ammyy.com and connects to its IP, 176.56.184.37, in the port 80/tcp. This CC channel works fine and receives a string of binary data. The malware then connects to a group of four IPs: 95.211.191.142, 95.211.242.83, 88.198.6.56, 88.198.6.54. The IPs are connected in an endless loop. In each loop, each IP is contacted on port 443/tcp, 80/tcp and 8080/tcp, in that order mostly. These IPs could be sent in the binary answer from the first CC server or they could have been hardcoded in the malware binary. None of the looping servers were correctly executing the CC. This resulted in the malware not being able to activate of further receive orders.

Regarding capture 192-3, we observed; first, it resolves www.msftncsi.com and contacts the remote server in port 80. Then, it contacts the same IP (217.182.53.102) on port 443. It resolves the domain rl.ammyy.com, and contacts the domain in port 80 and exchange binary information. Then it contacts the IP 95.211.191.142, port 443 tcp (it sends some binary information). Repeating this process in an endless loop.

4.3 Capture Kover.B (ID 219)

The malware executed is probably a trojan called Kover.B, it was executed with proxy and without proxy. In both cases, the malware contacted the remote Command and Control server in port 443/TCP and established an encrypted channel.

Capture Kover.B (ID 219-1) with proxy
First, the malware tries to establishes a Command and Control channel with the IP 42.2.231.204 on port 80/TCP, with 149.80.126.178 on port 8080/TCP and with 119.116.67.233 on port 443/TCP. Most of them used an unknown encryption protocol. These connections failed because the proxy could not understand them. The only connection that is successful is to IP 42.2.231.204 because it used real HTTP. Then, the malware attempts to create C&C channels on port 8080/TCP and 443/TCP with several IP addresses. Most of these also failed for the same reason. When all these connections failed, the malware establish a C&C channel with the IP 185.117.72.90 on port 80/TCP using HTTP protocol. The malware contacted more than 50 IP addresses using port 80/TCP. Some of the domains are contacted but, after 10 seconds of non-response from the remote servers, it resets the connection. It establishes a CC SSL channel with the IP 168.150.126.63, and then it closes the connection. The it contacts IP 48.99.155.215 in port 443, and exchanges binary information.

Capture Kover.B (ID 219-2) without proxy
The malware starts by creating a C&C channel with the IP addresse **23.4.249.223** on port 80/TCP and port 443/TCP. In the first case, the response from the remote server

is *HTTP/1.1 301 Moved Permanently*. In the second case, it exchanges certificates and establishes an SSL CC channel, after that, it exchanges binary information.

Then, the infected computer establishes a connection to www.microsoft.com, which is odd for any malware. Since the Windows computer is configured not to update, this may be a result of the malware opening another process. There are also connections to www.download.windowsupdate.com.

At the same time, the malware tries to connect to at least 50 different IP addresses on port 443/TCP. Most of them do not respond and after of 10 seconds of inactivity and it resets the connection. Eventually it establishes an SSL communication with the IP **198.144.30.128** on port 443/TCP, related with the domain *store.korfx.com*. It is clear that these IP are hardcoded in the malware.

The malware contacted different IP addresses on port 80/TCP trying to reach different C&C servers. However, most of them were not working. Only 4 IP addresses were contacted on port 443/TCP as C&C servers. It sends Sync packets up to ~5 packets per second to several IPs in ports 80, 8080 and 443

4.4 Capture Trojanized BitTorrent with Open Candy (ID 208)

This malware is probable a BitTorrent client that was *trojanized* with the adware Open Candy. It was executed twice; with and without proxy. In the two scenarios, the malware contacted the port 443/TCP and established a communication with a remote server. However, it was not an encrypted communication.

This capture is shown as an example of how there may be situations where the use of MITM proxy does not have an impact in the traffic. Since this malware does not have a Command and Control channel, it did not have any issue with the proxy.

Capture Trojanized BitTorrent with Open Candy (ID 208) with proxy
During this execution, the malware resolved different domains (i-50.b-000.xyz.bench.utorrent.com, i-21.b-42606.ut.bench.utorrent.com and bittorrent.vo.llnwd.net) related with the operation of BitTorrent. Most of the connections and updates are done using port 80/TCP. Some IP addresses are also contacted using UDP packets, due to the operation of the P2P protocol. Finally, it contacts other domains (previously contacted using the port 80), using the port 443/TCP. This process is repeated several times during the capture.

Capture Trojanized BitTorrent with Open Candy (ID 208) without proxy
This capture has the same behavior as the previous capture (208-1, with proxy). It contacts some domains on port 80, then it establishes an SSL communication with those IPs on port 443.

4.5 Discussion

After executing the malware with and without the web TLS proxy interception we discover a very important characteristic of the problem studied: the behavior of most malware changed when it was executed with a proxy. The most important findings are

three. First, while using the interceptor proxy, certain amount of malware, was not able to communicate to Internet at all, because the protocol used on port 443/TCP was not TLS. Therefore, the proxy refused to establish the connection. This forced the malware to new actions. Second, when the malware didn't achieve the connection to its remote servers, two new behaviors were observed: (1) the malware tried to reconnect continually to its C&C server, generating huge noise in the network, and (2) the malware seek another way to connect (choosing a different port, or looking for others servers). An example of this difference can be seen in Figure 2 and Figure 3. In particular, Figure 2 describes the behavior of the malware when was captured with mitmproxy and Figure 3 illustrates the behavior of the same malware without using a proxy. Third, some malware running with the web interceptor was not able to establish the C&C channel, while the capture without the web interceptor was able to do it. In these cases, the intercepted malware sent more than three times the amount of traffic compared with the not intercepted malware.

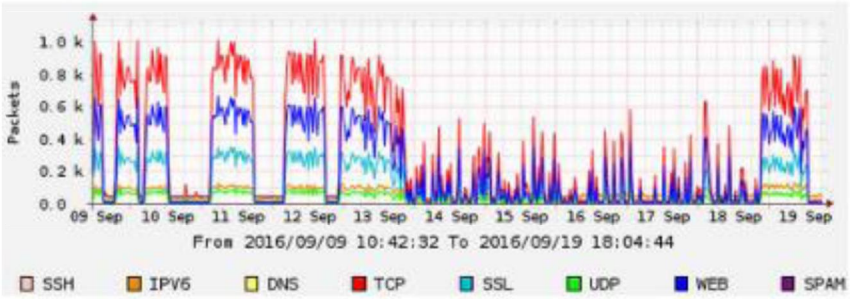

Figure 2. Malware captured with mitmproxy

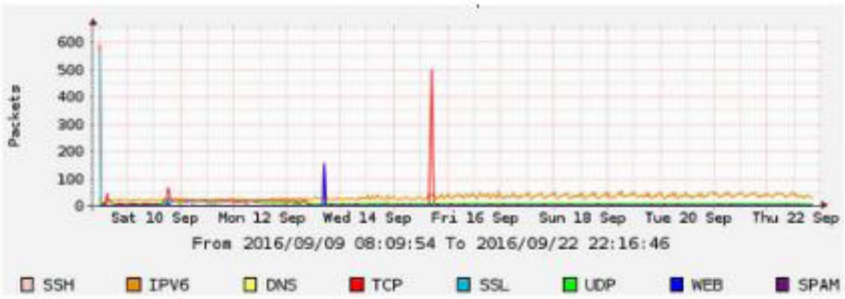

Figure 3. Malware captured without mitmproxy

5. Conclusion and future work

The analysis of the behavior of malware inside an encrypted connection is difficult. For this reason, we consider our malware dataset and analysis of encrypted behavior a step forward in this topic. There are two main conclusions regarding the impact of using a proxy for malware analysis. First, we observed a large amount of malware using a custom protocol on ports reserved for the HTTPs/HTTP protocols (ports 80,

8080 and 443). Blocking these connections forced the malware to generate new behaviors. Second, we noticed that malware's behavior can change in undefined ways when an intercepting proxy is used. Therefore, the implementation of a proxy should be carefully considered to detect malware in the network.

As future work, we will analyze more in detail other features in the captures. First, to detect which malware we are executing and second, to identify which version of the protocol the malware is using. When we executed the malware samples, we were provided of information from virus total and hybrid analysis. Even though this site is the more important virus database, it provides information from the possible name of the malware. For that reason, we consider that it is essential to verify which malware are we really executing. Second, we compared the differences between the malware that was using encrypted communication and not, but we did not analyze the version of the encryption protocol it was using.

References

1. O'Neill, M., Ruoti, S., Seamons, K., & Zappala, D. (2017). TLS Inspection: How Often and Who Cares? *IEEE Internet Computing*, *21*(3), 22–29. http://doi.org/10.1109/MIC.2017.58
2. Xavier de Carné de Carnavalet and Mohammad Mannan. Killed by Proxy: Analyzing Client-end TLS Interception Software. 21-24 February 2016, San Diego, CA, USA. Copyright 2016 Internet Society, ISBN 1-891562-41-X. http://dx.doi.org/10.14722/ndss.2016.2337
3. Lokoč J., Kohout J., Čech P., Skopal T., Pevný T. (2016) k-NN Classification of Malware in HTTPS Traffic Using the Metric Space Approach. In: Chau M., Wang G., Chen H. (eds) Intelligence and Security Informatics. PAISI 2016. Lecture Notes in Computer Science, vol 9650. Springer, Cham
4. František Střasák. Detection of HTTPS Malware Traffic. Open Informatics, Computer and Information Science. May 2017. https://dspace.cvut.cz/bitstream/handle/10467/68528/F3-BP-2017-Strasak-Frantisek-strasak_thesis_2017.pdf?sequence=-1
5. Anderson, B., Paul, S., & McGrew, D. (2016). Deciphering Malware's use of TLS (without Decryption). Retrieved from http://arxiv.org/abs/1607.01639
6. Blake Anderson. Hiding in Plain Sight: Malware's Use of TLS and Encryption. January 25, 2016. http://blogs.cisco.com/security/malwares-use-of-tls-and-encryption
7. Blake Anderson, David McGrew, Alison Kendler. Cisco Systems, Inc. Classifying Encrypted Traffic with TLS-Aware Telemetry. January 2016 . http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=449962
8. Stratosphere Dataset. https://stratosphereips.org/category/dataset.html
9. Nomad Project. https://stratosphereips.org/category/Nomad.html