Malicious SSL Certificate Detection: A Step Towards Advanced Persistent Threat Defence

Ibrahim Ghafir¹², Vaclav Prenosil¹, Mohammad Hammoudeh², Liangxiu Han², Umar Raza²
¹ Faculty of Informatics, Masaryk University, Brno, Czech Republic
² Faculty of Science & Engineering, Manchester Metropolitan University, Manchester, UK
ghafir@mail.muni.cz, prenosil@fi.muni.cz, M.Hammoudeh@mmu.ac.uk, l.han@mmu.ac.uk, u.raza@mmu.ac.uk

ABSTRACT
Advanced Persistent Threat (APT) is one of the most serious types of cyber attacks, which is a new and more complex version of multi-step attack. Within the APT life cycle, continuous communication between infected hosts and Command and Control (C&C) servers is maintained to instruct and guide the compromised machines. These communications are usually protected by Secure Sockets Layer (SSL) encryption, making it difficult to identify if the traffic directed to sites is malicious. This paper presents a Malicious SSL certificate Detection (MSSLD) module, which aims at detecting the APT C&C communications based on a blacklist of malicious SSL certificates. This blacklist consists of two forms of SSL certificates, the SHA1 fingerprints and the serial & subject, that are associated with malware and malicious activities. In this detection module, the network traffic is processed and all secure connections are filtered. The SSL certificate of each secure connection is then matched with the SSL certificate blacklist. This module was experimentally evaluated and the results show successful detection of malicious SSL certificates.

CSCS CONCEPTS
•Security and privacy → Intrusion detection systems; Network security; •Networks → Network monitoring;

KEYWORDS
Cyber attacks, malware, advanced persistent threat, malicious SSL certificate, intrusion detection system.

ACM Reference format:

1 INTRODUCTION
Cyber attacks refer to intentional activities against software, hardware or data in computer networks or systems. These activities may degrade, disrupt, destroy, or deny access to legitimate users. Many intelligence agencies and governments’ militaries are actively getting ready to launch or block cyber attacks, maybe in conjunction with traditional attacks or counter-attacks. Cyber exploitation is another expression, which refers to intelligence-gathering rather than devastating actions [17]. The goal is to get information and data without getting caught or being detected. As the number of ubiquitous devices grow [1, 2, 33–35, 42, 47, 50], the number of the security threats with implication for the general public increases. Unfortunately, new technologies, such as the Internet of Things, comes with new set of security threat that needs to be addressed.

Over the last decade, malware and botnets [28, 31] have increased to become a key reason of the majority of the (Distributed) Denial-of-Service (DOS) activities [38], direct attacks [4], spear phishing [22] and scanning [51], which takes place through the Internet. Botnets are networks formed by “enslaving” host computers, called bots (derived from the word robot), that are controlled by one or more attackers, called botmasters, with the intention of performing malicious activities [5]. In other words, bots are malicious codes running on host computers that allow botmasters to control the host computers remotely and make them perform various actions [15]. It has been noticed that there is a change in motivation, from curiosity and fame seeking/excitement-seeking to illegal financial gain, this has been marked by a rising sophistication in the evolution of malicious software [16]. Moreover, the availability of easy-to-use tool kits to build malware will probably keep malware a threat to consumers, businesses and governments in the foreseeable future. Furthermore, network monitoring approaches [52] have become more and more important in modern complicated networks.

One of the most serious types of cyber attacks is the Advanced Persistent Threat (APT) [20], which is targeting a specific organisation and it is performed through several steps. The main aim of APT is espionage and then data exfiltration. Therefore, APT is considered as a new and more complex version of multi-step attack [36]. These APTs form a problem for current detection methods [19] as they use advanced techniques like social engineering [27] and make use of unknown vulnerabilities. Moreover, the economic damages due to a successful APT attack can be very expensive. The expected cost of attacks is the major motivation for the investments in intrusion detection and prevention systems [46]. APTs are currently one of the most serious threats to the companies and governments [54].

A novel approach for APT detection is proposed in [26]. The suggested system undergoes two main phases, the first one detects eight techniques commonly used in APT life cycle. For that purpose, eight detection modules are presented, which are disguised exe file detection [18], malicious file hash detection [25], malicious domain name detection [24], malicious IP address detection [23], malicious

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICFNDS ’17, Cambridge, United Kingdom
© 2017 ACM. 978-1-4503-4844-7/17/07 ... $15.00
DOI: 10.1145/3102304.3102331
SSL certificate detection, domain flux detection [21], scan detection, and Tor connection detection [30]. The second phase includes a correlation framework to link the outputs of the detection modules.

Within the APT life cycle, continuous communication between the infected hosts and the C&C servers should be preserved to instruct and guide the compromised machines. These communications are usually protected by Secure Sockets Layer (SSL) encryption, making it difficult to identify if the traffic directed to sites is malicious. This paper presents the malicious SSL certificate detection (MSSLD) module, which aims at detecting the APT C&C communications based on a blacklist of malicious SSL certificates. This blacklist consists of two forms of SSL certificates, the SHA1 fingerprints and the serial & subject, that are associated with malware and malicious activities. In this detection module, the network traffic is processed and all secure connections are filtered. The SSL certificate of each secure connection is then matched with the SSL certificate blacklist.

The remainder of this paper is organized as follows. Section 2 presents the related work to APT detection. The malicious SSL certificate detection module and its algorithms are explained in Section 3. Section 4 shows the evaluation results and Section 5 concludes the paper.

2 RELATED WORK

A classification model for APT detection is presented in [14]. This model is built on machine learning algorithms. First, the legitimate traffic for normal users is analysed aiming to extract the CPU usage, memory usage, open ports and number of files in the system32 folder. A piece of malware, which is previously used for the APT attack, is injected into the network and the four features are extracted. The dataset of benign and malicious features are used to train the detection model using different machine learning algorithms.

TerminAPTor, an APT detector, is described in [12]. This detector uses information flow tracking to find the links between the elementary attacks, which are triggered within the APT life cycle. TerminAPTor depends on an agent, which can be a standard intrusion detection system, to detect those elementary attacks. A statistical APT detector, similar to TerminAPTor detector, is developed in [48]. This system considers that APT undergoes five states which are delivery, exploit, installation, C&C and actions; and several activities are taken in each state. The generated events in each state are correlated in a statistical manner.

An APT detection system based on C&C domains detection is introduced in [53]. This work analyses the C&C communication based on the observation that the access to C&C domains is independent, while the access to legal domains is correlated.

An approach for APT detection based on spear phishing detection is explored in [13]. This approach depends on mathematical and computational analysis to filter spam emails. Tokens, which are considered as a group of words and characters such as (click here, free, Viagra, replica), should be defined for the detection algorithm to separate legitimate and spam emails.

An active-learning-based framework for malicious PDFs detection is suggested in [40]. These malicious PDFs can be used in the early steps of APT to get the point of entry.

An approach based on Data Leakage Prevention (DLP) is proposed in [49]. This approach focuses on detecting the last step of APT which is the data exfiltration. A DLP algorithm is used to process the data traffic to detect data leaks and generate "fingerprints" according to the features of the leak.

A context-based framework for APT detection is explained in [32]. This framework is based on modelling APT as an attack pyramid in which the top of the pyramid represents the attack goal, and the lateral planes indicates the environments involved in the APT life cycle.

An in-depth analysis of Duqu is presented in [7]. A European organisation was targeted by attackers using the Duqu malware to steal data. The authors propose the Duqu detector toolkit, which consists of six investigation tools developed to detect the Duqu malware involved in the APT attacks.

With regards to the processing of multiple streams of events, IBM suggests a conceptual model for event processing in [39]. It describes the basic requirements to design an efficient correlation system. The work presented in [10] is based on finite state machines and uses a query language for event processing. Both systems can process the events in real time, a key limitation shared by both approaches is that they can not detect so called low & slow attacks, which take place over an extended time period.

Finally, APT detection systems face serious shortcomings in achieving real time detection [6], detecting all APT attack steps [6], balance between false positive and false negative rates [7], and correlating of events spanning over a long period of time [10, 39]. To address those weaknesses, a new approach for APT detection has been presented in [26], and this paper is a step towards developing the proposed system.

3 MALICIOUS SSL CERTIFICATE DETECTION (MSSLD)

APT C&C communications are usually protected by Secure Sockets Layer (SSL) encryption, which makes it difficult to identify malicious traffic. MSSLD aims at detecting C&C communications based on a blacklist of malicious SSL certificates [3, 37]. This blacklist consists of two forms of SSL certificates, the SHA1 fingerprints and the serial & subject, which are associated with malware and malicious activities. As shown in Figure 1, the network traffic is processed and all secure connections are filtered. The SSL certificate of each secure connection is then matched with the SSL certificate blacklist.

The MSSLD module is implemented on top of the Bro [29, 43] passive, open-source network traffic analyser. It is primarily a security monitor, which inspects all traffic on a link in depth for signs of suspicious activity. The most immediate benefit gained from deploying Bro is an extensive set of log files, which record a network’s activity in high-level terms. These logs include not only a comprehensive record of every connection seen on the wire, but also application-layer transcripts such as, e.g., all HTTP sessions with their requested URLs, key headers, MIME types, and server responses; DNS requests with replies; and much more. Bro event engine reduces the incoming packet stream into a series of higher-level events, more than 300 events. These events reflect network activity in policy-neutral terms, i.e., they describe what has been
seen, but not why, or whether it is significant. The MSSLD module consumes and handles some of Bro events, explained later in Section 3.1; all connection information (such as timestamp, src_ip, src_port, dest_ip, dest_port) can be extracted from those events.

This detection module runs through two algorithms, mentioned further in Section 3.1, the first one is intelligence-based MSSLD and the second algorithm is event-based MSSLD. The intelligence-based MSSLD makes use of the Bro Intelligence Framework [11]. This framework enables attack detection modules to consume data from different data sources and make it available for matching.

MSSLD is a blacklist-based detection module in which the malicious SSL certificates blacklists are automatically updated based on different intelligence feeds at once. The automatic update runs parallel with the module process and there is no need to stop or restart MSSLD. This parallel-running feature allows a continuous live monitoring of the network traffic and supports real time detection. Based on the MSSLD algorithm, two automatic update mechanisms have been applied. Figure 2 shows the automatic update of the blacklists used by intelligence-based MSSLD. The user crontab file is configured to run blacklist_update.sh each day at 3:00 am, this shell script connects through the Internet to the data source servers and downloads updated blacklists of malicious SSL certificate hashes into a new blacklist.intel text file. This text file is connected to the Intelligence Framework, which consumes it as described in Section 3.1.

Figure 2: Automatic update of the blacklist used by intelligence-based MSSLD.

Figure 3 shows the automatic update of the blacklists used by the event-based MSSLD. The updated blacklist is downloaded into ssl_blacklist.txt text file. The Input Framework [44], built in Bro, enables MSSLD to use that text file as an input. The Input Framework reads ssl_blacklist.txt file into bad_ssl group, which is used by event-based MSSLD.

As an output of the MSSLD module, in case of a malicious SSL certificate is detected, a corresponding event (ssl_alert) is generated. This event is to be used for alert correlation [8]. Additionally, an alert email is sent to the Request Tracker (RT) [9], where the network security team can perform additional forensics and respond to the triggerer alert. It is assumed that the network security team responds to the generated alert within 24 hours, therefore, MSSLD suppresses all the same alerts, the same alert is the one which has the same infected host and the same malicious SSL certificate, into one alert per day, so no repeated alert emails sent the network security team. Moreover, this alert suppression reduces the computational cost of the alert correlation framework. To this end, after an alert is generated, this module adds the triggered alert into a specific table, the t_suppress_ssl_alert, where it stays for one day to ensure that the module does not generate the same alert within the next 24 hours. When a malicious SSL certificate is detected, and before an alert is generated, MSSLD checks the t_suppress_ssl_alert table in order to conclude if the same alert has been generated during the previous day, if so, the alert is ignored. Along with generating a new alert, information regarding the alert and the malicious certificate is witten into a specific log, i.e., blacklist_detection_ssl.log, to keep a historical record of the monitored network.

3.1 MSSLD Algorithms

As the blacklist consists of two forms of malicious SSL certificates (SHA1 fingerprints and serial & subject), two methods are followed for malicious SSL certificate detection. The first one is intelligence-based MSSLD, shown in Algorithm 1, and the second method is event-based MSSLD, shown in Algorithm 2.

In the intelligence-based MSSLD, the Bro Intelligence Framework is used and configured to monitor all secure connections SSL certificates’ hashes. This framework is connected to the blacklist.intel file, which contains the SSL certificate blacklist. After extracting all secure connections traffic, SSL certificates hashes are passed to the intelligence framework to be checked against the intelligence data set blacklist.intel. When a match with any indicator_type of the intelligence data is found, the intelligence framework generates an Intel:match event. Through this event, if the indicator_type is CERT_HASH, it means this connection has
Algorithm 1: Implementation pseudo-code of intelligence-based MSSLD

1. Get malicious SSL certificates hashes blacklist (blacklist.intel)
2. Filter secure connections traffic
3. Extract SSL certificate hash
4. Send SSL certificate hash to Bro Intelligence Framework
5. if SSL certificate hash is in blacklist.intel then
6. if the connection source IP belongs to the monitored network then
7. if the same ssl_alert had not been generated over the last day then
8. Generate an event (ssl_alert)
9. Write ssl_alert into blacklist_detection_ssl_log
10. Send an alert email to RT
11. Suppress the same ssl_alert over the next day
12. end if
13. else if the connection destination IP belongs to the monitored network then
14. if the same ssl_alert had not been generated over the last day then
15. Generate an event (ssl_alert)
16. Write ssl_alert into blacklist_detection_ssl_log
17. Send an alert email to RT
18. Suppress the same ssl_alert over the next day
19. end if
20. else
21. goto End
22. end if
23. else
24. goto End
25. end if
26. End

Algorithm 2: Implementation pseudo-code of event-based MSSLD

1. Get malicious SSL certificates [serials and subjects] (bad_ssl group)
2. Filter secure connections traffic
3. Get x509.certificate event
4. Extract SSL certificate [serial and subject]
5. if SSL certificate [serial and subject] is in bad_ssl then
6. if the connection source IP belongs to the monitored network then
7. if the same ssl_alert had not been generated over the last day then
8. Generate an event (ssl_alert)
9. Write ssl_alert into blacklist_detection_ssl_log
10. Send an alert email to RT
11. Suppress the same ssl_alert over the next day
12. end if
13. else if the connection destination IP belongs to the monitored network then
14. if the same ssl_alert had not been generated over the last day then
15. Generate an event (ssl_alert)
16. Write ssl_alert into blacklist_detection_ssl_log
17. Send an alert email to RT
18. Suppress the same ssl_alert over the next day
19. end if
20. else
21. goto End
22. end if
23. else
24. goto End
25. end if
26. End

a malicious SSL certificate. Next, both connection sides, source and destination IP addresses, are checked through the is_local_addr function to check if the connection is established to or from the monitored network. To avoid raising the same alert within the same day, the t1_suppress_ssl_alert table is checked to ensure that it does not contain the same detected [host IP address, SSL certificate hash] set.

MSSLD then generates ssl_alert event, writes the malicious connection information into a specific log blacklist_detection_ssl_log, sends an alert email regarding the malicious SSL certificate detection to RT and adds the current detected set [host IP address, SSL certificate hash] into t1_suppress_ssl_alert table. The written information into blacklist_detection_ssl_log is:

timestamp = s$conn$start_time
alert_type = "ssl_alert"
connection = s$conn$id
infected_host = s$conn$ids$orig$h
malicious_ssl = s$indicator

In the event-based MSSLD, the network traffic is processed and filtered into secure connections traffic, and then x509.certificate event can be generated for encountered X509 certificates [45]. Through this event, the serial and subject of the X509 certificate are checked for the certificate presence in the bad_ssl group. This group contains many of serials and subjects of malicious X509 certificates. If a match is found, the module should determine if the connection is established to or from one of the monitored network hosts; accordingly, both the source and destination IP addresses are checked through is_local_addr function. Before an ssl_alert is raised, the t2_suppress_ssl_alert table is to be checked to ensure that the same alert was not raised previously during the same day.

As in the previous intelligence-based method, MSSLD generates an ssl_alert event, writes the malicious connection information into a specific log, i.e., the blacklist_detection_ssl_log, sends an alert email regarding the malicious SSL certificate detection to RT and adds the current detected set [host IP address, SSL certificate hash] into t2_suppress_ssl_alert table.

4 EVALUATION RESULTS

To evaluate the MSSLD module, a virtual Internet-connected network was built, malware samples were injected into the virtual network, the network traffic was recorded into pcap files, and then the MSSLD module was applied on those pcap files.

As illustrated in Figure 4, two Windows virtual machines were connected to a physical consumer-grade router, which provided
connection to the Internet. The virtual machines behaved as physical computers in a home network and were able to communicate with each other. The Virtual machines traffic was recorded into two separate pcap files using the nictrace VirtualBox functionality [41]. Because no software besides the operating system and the malware was installed on the virtual machines and the operating system updates were disabled, the majority of the captured traffic was initiated by the installed malware. Moreover, it becomes easy to establish the ground truth.

**Figure 4: Topology of the implemented virtual network.**

Two malware samples were injected independently into the virtual network for 5 minutes each. The first one is the Trojan.Win32.Inject.shgq, also known as TorrentLocker, which has MD5 hash value of aabe284e6e1e1f29694d7a96e1355a99. The second injected malware sample is the Trojan.Win32.Staser.bazr malware, which has MD5 hash value of e161a4d2716eb3352d3b2e5d03c. The C&C servers for these two malware uses SSL certificates for communication over https. When the MSSLD module was applied on the captured pcap files, it successfully detected the malicious traffic as shown in Figure 5.

**Figure 5: Part of a log produced by the MSSLD module.**

## 5 CONCLUSION AND FUTURE WORK

This paper presents the malicious SSL certificate detection (MSSLD) module, which aims at detecting theAPT C&C communications based on a blacklist of malicious SSL certificates. This blacklist consists of two forms of SSL certificates, the SHA1 fingerprints and the serial & subject, that are associated with malware and malicious activities. This module, processes the network traffic and filters all secure connections. The SSL certificate of each secure connection is then matched with the SSL certificate blacklist.

For future work, the output of this module will be correlated with the outputs of other detection modules, developed to detect commonly used techniques over the APT life cycle, to raise an alert on APT detection.

## REFERENCES


