Design for Network Attack Forensic System Based on HTTP Evasive Behavior

Wenhao Liu, Haiqing Pan, Gang Xiong, Zigang Cao and Zhen Li

Abstract The network traffic generated by humans and various devices is one of the most important data sources in network forensics. The main challenge in investigating and collecting evidence in network traffic is handling the huge amounts of data streams caused by the rapid growth of network bandwidth and applications, as well as preserving the useful information for further analysis. HTTP, as the most popular protocol on the Internet, is usually exploited to carry malware and evasive attacks besides the normal services. In this paper, we study how malware and network attacks in real-world exploit HTTP to hide their malicious activities and present an Evasive Network Attack Forensic System (ENAFS), which is able to effectively discover evasive network attacks on HTTP and integrally draw attack the samples and their metadata for further analysis. We believe that our work will benefit the research in the network forensics field in the future.

Keywords Network forensics · HTTP · Evasive attack · Abnormal behavior

1 Introduction

Nowadays, HTTP holds a quite large portion of the network traffic volume. Many attackers and malware exploit HTTP to hide their malicious activities due to its popularity and flexibility [1]. A typical kind of behavior is that one Content-Type is declared in the HTTP header while in fact the actual data is of another different type in order to evade security inspections. Such kind of behavior has been observed in some famous malware, such as Zeus, Torpig, Bredolab [2]. Some Advanced Persistent Threat (APT) attacks, like APT30 [3] and APT Operation Poisoned Helmand [4], also used this trick to bypass detection. Therefore, designing a forensic system to record and analyze the malware and APT attacks behind the

W. Liu · H. Pan · G. Xiong · Z. Cao · Z. Li (✉)
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
e-mail: lizhen@iie.ac.cn

J.J. (Jong Hyuk) Park et al. (eds.), *Advanced Multimedia and Ubiquitous Engineering*, Lecture Notes in Electrical Engineering 393,
DOI 10.1007/978-981-10-1536-6_63

evasive behaviors are very valuable for discovering potential network threat and enhancing the network security.

However, the Content-Type mismatches in HTTP take 35 % of the total HTTP volume [5], and most of them are caused by innocent configuration mistakes. Therefore, it is not a sensible way to simply detect the mismatch by a general rule since there must be a lot of false positives. Meanwhile, the traditional rule-based intrusion detection systems (IDS), e.g. Snort, can only detect limited known Content-Type mismatch attacks based on their rules with few clew in logs which is helpless to investigate and trace these attacks.

To mitigate the problem, we design Evasive Network Attack Forensic System (ENAFS) to record the HTTP Content-Type mismatch data and automatically analyze the latent evasion behaviors. We've deployed the proposed system on a real-world network. The system processed 1.2 billion HTTP requests per day, found 166 million mismatch data and recorded thousands of latent evasive attack samples.

The main contributions of this paper are as follows:

- A novel network attack traffic forensic system is proposed and implemented to record traffic samples and detect evasive network attacks.
- The proposed system is deployed in the real-world network. From 1607 mismatch types recorded by the system, several specific mismatch types that network attacks tend to use to hide their activities have been found.
- Some previously-unseen malware have been detected in the real-world network traffic. These malware activities and samples have been preserved by the system for further analysis.

2 Related Work

Mukkamala and Sung [6] use two artificial intelligence techniques, (Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), to identify significant features for offline network intrusion analysis, and test them with 1999 DARPA intrusion data. However, this technique is not suitable for real-time network traffic detection. Kaushik and Joshi [7] propose a network forensic system for ICMP attacks and just focus on some specific attacks on ICMP protocol. This system can only detect known attacks since it is based on rules. Cohen [8] proposes a network forensic framework to reassemble streams, parse HTTP protocol and list the traffic content visually. Pilli and Joshi introduce and compare 11 Network Forensic Analysis tools [9], and mention two tools, Infinistream and OmniPeek, which have real-time analysis capabilities. Our work is also based on the real-time traffic analysis. But we further research on the mismatch scenarios and apply it in the proposed ENAFS to detect and trace the latent evasion attacks.

3 Our Approach

In this section, the ENAFS is descripted in detail, including suspicious HTTP traffic collection, sample detection, and attack tracing. The framework of ENAFS is shown in Fig. 1 and explained below.

3.1 Traffic Collection

ENAFS aims at discovering the latent evasive attack using a fake HTTP content-type header. Therefore, we first need to find out the real HTTP payload type and compare it with the HTTP content-type header.

To achieve this goal, we extract HTTP packets based on PF_RING and use *libmagic* (file type determination library in GNU/Linux) to identify the HTTP payload MIME type and perform a string comparison between libmagic's results and content-type header.

In most of the cases, the comparison results are correct, but sometimes they are not accurate when meeting the following situations.

- Network transmission errors may cause libmagic fail to detect the HTTP payload header and make it return a general type application/octet-stream.
- Libmagic returns the same MIME type with the Content-Type header, but in different description. With Microsoft Application, for example, libmagic

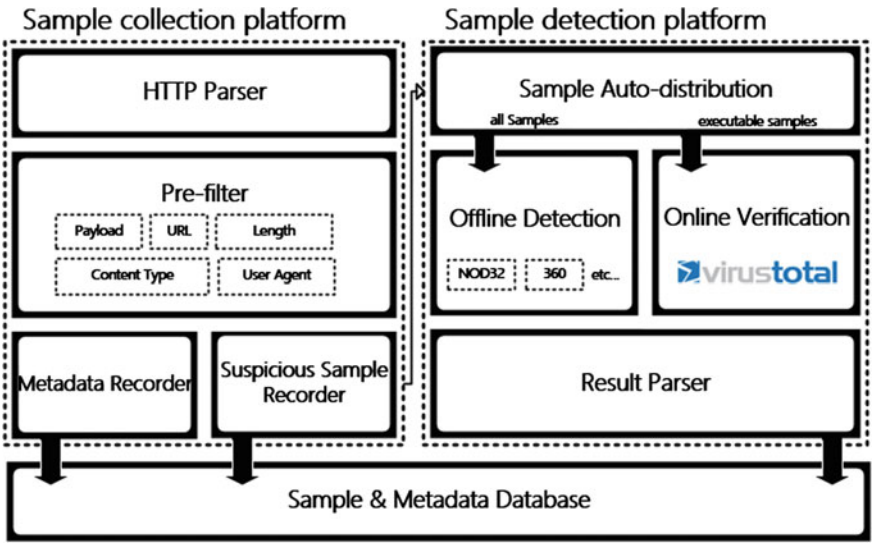

Fig. 1 Network attack forensic system framework

describe it as the type "application/x-dosexec", while Content-Type header describe it as "application/x-msdownload".
- In the Text, Media, and Image classes, libmagic and the Content-Type header agreed on the general category of the type (e.g., image) but disagree on the actual file format (JPEG vs. PNG).
- HTTP payload with *gzip* compression may cause the result inaccurate.

In order to reduce false positive in comparison, we decompress the payload zipped with *gzip* protocol and create a hash map to handle the inaccurate comparison results to make sure that the results are performing as expected.

After comparison, ENAFS will rerecord the metadata of content mismatched HTTP sessions, including source and destination IP address and port, URI, User-Agent and Server field in the HTTP header. In addition, the complete payload sample will also be reassembled and stored with a hash value for further detection.

3.2 Sample Detection and Attack Tracing

Two methods are implemented to detect the suspicious malicious samples. One is offline scanning and the other is online verification.

For offline scanning, five anti-virus software such as 360 Antivirus, Avira, ClamAV, ESET NOD32, and Kaspersky are used. 360 Antivirus is the most famous Antivirus in China, so we choose it to represent free Antivirus software in China. ClamAV is the most famous open source Antivirus software, and we choose it to be representative of open source Antivirus software. We also pick up three widely used and with well performance Antivirus, Avira Antivirus, ESET NOD32 Antivirus, and Kaspersky Antivirus to our detection system. The information of these five Antivirus [10] is shown in Table 1.

These five Antivirus are deployed separately in separate virtual machines, each of them receives suspicious samples from Sample Auto-distribution Module, and executes scanning automatically by the scripts. After scanning, results will be transferred to Scan Result Parsing module. Scan Result Parsing module analyzes scan result files received from Antivirus, and then store the scan result and its metadata into the database.

Table 1 Information of antivirus used by ENAFS sample detection platform

Antivirus	Signature-based scanning	Heuristic scanning	Open source
Avira antivirus	Yes	Yes	No
360 antivirus	Yes	Yes	No
ClamAV	No	Yes	Yes
ESET NOD32 antivirus	Yes	Yes	No
Kaspersky antivirus	Yes	Yes	No

For online verification, we choose VirusTotal, a well-known online scan service which aggregates 55 antivirus products and 61 online scan engines [11] to check for viruses. The offline detection has a better performance while the online verification has a higher recall rate but is relatively slow. Hence, the offline detection is used to scan all the samples that system recorded but only highly suspicious samples will be uploaded for online verification. Currently the system only uploads all the executable samples for online verification.

After sample detection, analysts can take advantage of the comprehensive and well-organized information in database to trace back the attacks. For example, the *URI* and *IP* address can be used to locate the victim and C&C server, *User-agent* field can tell us victim's device (e.g. Mobile or PC) and OS type. *Sample hash* can be used to reveal the relationship of the same attack from different sources.

4 Result Analysis

An offline data set from third part is used to verify our detecting system. The VRT Lab [2] has provided some malicious malware traffic contains evasive attack activities which are suitable for use as benchmark of ENAFS. As is presented in Table 2, our system holds a highly reliable result for detecting malicious malware in third-party.

ENAFS was also used in our network board to detect malicious attack activities. We deployed ENAFS on the boarding entry of CERNET network for 7 days, which serves China's research institution and universities. As a result, ENAFS processed over 8.4 billion HTTP sessions and found out more than 110 million mismatch instances, covering 1607 different kinds of type mismatches. All these mismatch HTTP metadata and payload samples were stored in the attack tracking database. After scanning and analyzing these samples, two types of evasive attacks had been found. One is that malware use image headers or text to hide its executable attribute. The other is that malicious scripts, such as webshell, claims as image to cheat the server-side format verification.

Finally, we tried to trace back the origin of a typical evasive attack recorded by the system. This attack sample was captured on a lottery website. It is an executable

Table 2 ENAFS offline data set detection results

Malware	Attack type	Content-type, Payload-type	Unique samples	Detection rate
Zeus	botnet	application/x-dosexec, image/jpg	40	100 %
Bredolab	botnet	application/x-dosexec, text/php	10	100 %
Tro-downloader	evasive Trojan downloader	application/x-dosexec, image/x-icon	120	100 %

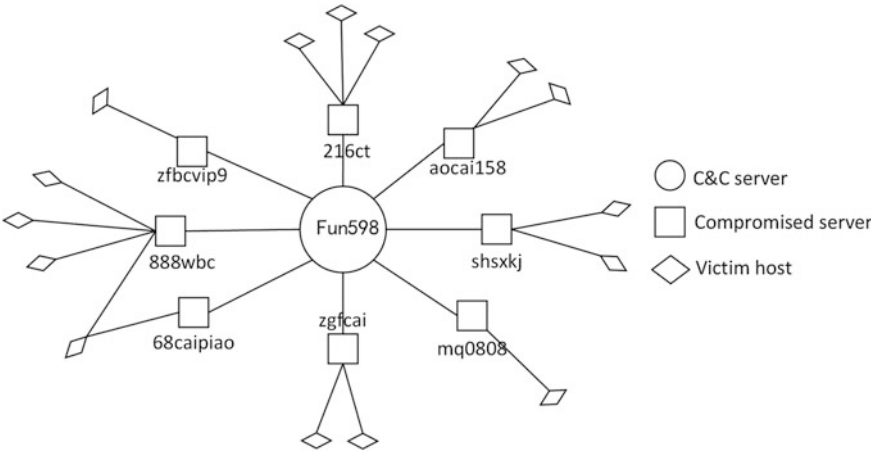

Fig. 2 Evasive network attack tracing graph

file named as *favicon.ico*, which is a common name for a web server, to disguise its real malicious purpose. When victims visit the lottery website, it will trigger drive-by downloads and infect the host. The system also captured other similar instances for *favicon.ico*. There are 15 samples in 8 websites trying to evade detection in such way, and all of them come from illegal lottery sites. Among them, two samples are previously-unseen malware which passed the antivirus scanning. Further analysis showed that all of these malware have relation with the domain *fun598.com*. Figure 2 shows the source tracing graph of the attack generated by ENAFS records.

5 Conclusion and Future Work

In the paper, we developed ENAFS to discover evasive network attacks on HTTP, which is able to integrally preserve attack samples and their metadata from the live traffic. We use several techniques to reduce the false positive system generated. With the help of sample auto-detection platform integrated in ENAFS, analysts can quickly discover the evasive network attack behind the traffic, analyze its sample and trace back its source.

The proposed ENAFS was evaluated on the ISP network for 7 days, during which the system handled 8.4 billion HTTP sessions, downloaded and scanned 110 million samples. From the detection results, we found a typical attack and revealed its malicious activity through our system.

Currently we are analyzing samples that the system has collected, expecting to find more HTTP Evasive behavior and add these features into our system. In the future, we would like to extend our system to support more protocol.

Acknowledgments This work is supported by the National Science and Technology Support Program (No. 2012BAH46B02), Xinjiang Uygur Autonomous Region Science and Technology Project (No. 201230123) and the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA06030200).

References

1. Invernizzi L, Miskovic S, Torres R, Saha S, Lee SJ, Kruegel C, Vigna G (2014) Nazca: detecting malware distribution in large-scale networks. In: Proceedings of the ISOC network and distributed system security symposium
2. VRT Labs (2014) Content-type mismatch. https://labs.snort.org/papers/contentmi-smatch.html
3. FireEye (2015) APT30 and the mechanics of a long-running cyber espionage operation. https://www2.fireeye.com/rs/fireye/images/rpt-apt30.pdf
4. Operation Poisoned Helmand (2014) https://www.threatconnect.com/operation-poisoned-helmand/
5. Schneider F, Ager B, Maier G, Feldmann A, Uhlig S (2012) Pitfalls in HTTP traffic measurements and analysis. In: Passive and active measurement. Springer, Berlin, pp 242–251
6. Mukkamala S, Sung AH (2003) Identifying significant features for network forensic analysis using artificial intelligence techniques. Int J Digital Evid IJDE 3
7. Kaushik AK, Joshi RC (2010) Network forensic system for ICMP attacks. Int J Comput Appl 2(3):14–21
8. Cohen M (2008) PyFlag—an advanced network forensic framework. Digital Inv 5:112–120
9. Pilli ES, Joshi RC, Niyogi R (2010) Network forensic frameworks: survey and research challenges. Digital Inv 7(1/2):14–27
10. Comparison of antivirus software (2016) https://en.wikipedia.org/wiki/Comparison_of_antivirus_software
11. VirusTotal (2015) https://en.wikipedia.org/wiki/Virus-Total