Poster: A Low-cost Detection Scheme on Fast-flux Malware Distribution

Haiqing Pan, Wenhao Liu, Gaopeng Gou, Zigang Cao, Zhen Li, Gang Xiong
Institute of Information Engineering, Chinese Academy of Sciences
Beijing, China
{panhaiqing, liuwenhao, gougaopeng, caozigang, lizhen, xionggang }@iie.ac.cn

***Abstract*—Malware is one of the most severe crucial security threats on the Internet. Many malware authors frequently change their malware distribution domains and URLs to avoid IDS (Intrusion Detection Systems) detection, and URLs become invalid shortly, which we name the malware distribution phenomenon fast-flux. We proposed a scheme named LDSM which could be** able **to detect fast-flux malware distribution with low cost. We have deployed our detection scheme in an ISP of CSTNET (China Science and Technology Network) for a month, and the experimental results demonstrate that LDSM is able to accurately detect malware distribution with 89.57% true positives. LDSM also could detect unknown malware from traffic and provides an effective way to improve malware detection tools.**

Keywords—Fast-flux, Malware distribution, Malware detection.

I. INTRODUCTION

Malware is still one of the most severe crucial security threats on the Internet. A key challenge for attackers is to install their malware programs on as many victim machines as possible. When static domains and URLs are used to distribute malware, they can be easily detected or blocked by anti-virus software or browser URL blacklists such as Google Safe Browsing [1]. Many malware authors frequently change their malware distribution sites to deceive detection, which would lead the former URL to be invalid and cause fast-flux malware distribution phenomenon.

As Windows operating system is widely used and the malicious PE file occupies a large part of malware, We focus on PE malware distribution detection on network traffic.

There are some previous researches on PE malware distribution detection on network traffic. Rossow et al. [2] characterize some malware Downloader on network level. Ding et al. [3] capture malware Downloader, analyze Downloader binaries and use a graph-based algorithm to cluster them. Both Rossow and Ding just detect and analyze malware Downloader exclude generic malware distribution. Vadrevu et al. [4] measure the differences between malware and benign software on statistical features of HTTP traffic and propose a detection method based on statistical features. Invernizzi et al. [5] aggregate suspicious connections to a malicious neighborhood graph and identifies web requests related to malware distributions. [4] and [5] not only detect malware distribution, including Downloader effectively, but also could detect some fast-flux malware distribution.

All the methods mentioned above are complex and high resource consumption, therefore, we propose a low-cost and effective detection scheme to detect fast-flux malware distribution (LDSM). We have deployed our detection scheme in an ISP of CSTNET (China Science and Technology Network) for a month, and the experimental results showed that LDSM is able to accurately detect malware distributions with 89.57% true positive. LDSM can also detect unknown malware from traffic and provides an effective way to improve malware detection tools.

II. SYSTEM DESIGN

In this section, we introduce the pipelines of our system. LDSM consists of three main components, showed in Figure 1: (1) the *PE reconstruction* module, (2) the *download database*; (3) the *re-download* module. We will introduce how these components work in the following.

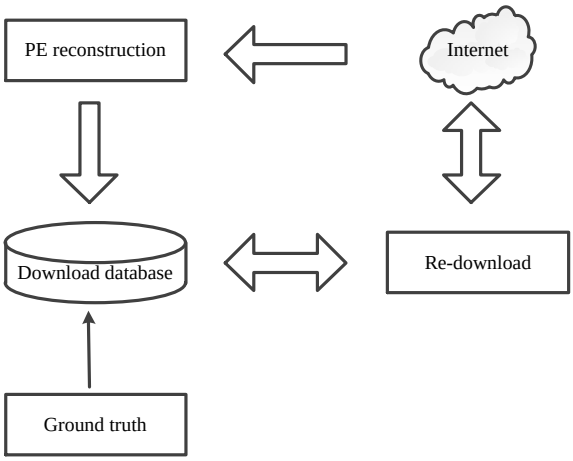

Fig. 1. LDSM system overview.

A. PE Reconstruction Module

The *PE reconstruction* module targets to extract Microsoft Windows portable executable (PE) files from web traffic. To achieve this goal, we extract HTTP packets based on RF-RING and use the ASCII string “MZ” (hexadecimal: 4D 5A) [6] to identify PE file.

It is known that updates of normal applications would also lead resource access to be failed, so in order to reduce false positive, we filter out PE files belonging to highly reputable companies, such as Microsoft, Facebook, Lenovo, Skype, Google and so on.

After confirming PE file and ensuring that it does not belong to high reputable companies, LDSM will reassemble and store the complete payload, which is used to be scanned by VirusTotal (VT) [7] and build ground truth, as well as record the metadata, including source and destination IP and port, URI, User-Agent and Server field in the HTTP header. Payload storing paths and the metadatas would be stored into *download database* by Python Scripts automatically. For the convenience of latter research, we added two items, fast_flux_tag and malicious_tag. Fast_flux_tag tags whether the PE file is fast-flux. Malicious_tag tags whether the PE file is malware provided by VT and this will be introduced detailly.

B. Re-download Module

Re-download module would automatically reassemble URL, and re-download the PE file on a timed basis by Scripts, one month after PE file is stored by the *PE reconstruction* module. LDSM will set fast_flux_tag to be '1', when it is failed to download the corresponding PE file.

We automatically remove those PE files and their metadata from *download database* by Python Scripts, whose item fast_flux_tag is not '1' and were stored into the database one month ago.

C. Download Database

The *download database* stores all information gathered by the *download reconstruction* module and *re-download* module.

For the purpose of building ground truth, we upload those PE files, whose fast_flux_tag is '1', to VT and set item malicious_tag to be '1' (means label the file as malware) if two or more antivirus software flagged the file as malware. All these operations is completely automatic accomplished by Python Scripts.

III. PRELIMINARY RESULTS

We have deployed our detection scheme in an ISP of CSTNET, whose bandwidth is 4Gbps, for a month. In our *download database*, there are 7,755 fast_flux_tag to be '1' and 6,946 malicious_tag to be '1'. The experimental results showed that LDSM is able to accurately detect malware distributions with 89.57% (6,946/7,755) true positive.

LDSM can also detect zero-day (unknown) malware [4] from traffic as VT initially classify the file as benign, and then label it as malware after one-month re-scan by VT. We found 52 zero-day MALWARE out of 809 PE file by VT re-scan.

We found that the Content-type of some zero-day malware is set to be image type, such as "image/png", "image/jpg" and "image/ico". This reflects that some malware not only churn its URL but also disguise to be a normal image file.

This kind of mismatch between Content-type and real MINE Type may be helpful for malware detection, thus we measure how many types of non-PE file, fast-flux malware disguise to be and the distributions. The measurement result is shown in TABLE 1, it is clear that malware prefers to disguise as image file..

TABLE 1. Content-type distribution of non-PE

Content-type	jpeg	gif	bmp	png	css	html	xml
sum	126	118	111	102	11	3	1

IV. CONCLUTION AND FUTURE WORK

A. Conclusion

In this paper, we propose a low-cost and effective detection scheme based on one-month re-download status to detect fast-flux malware. LDSM could also detect malware from network traffic and provides an effective way to improve malware detection tools.

We also took notice of mismatch between Content-type and real MINE Type, measured on Content-type distribution of non-PE file, and found that such kind of mismatch may be useful for detection.

B. Future Work

In the future work, we will firstly keep on going to capture more fast-flux PE file and expand the scale of our *download database*. Secondly analyze the relations between these fast-flux PE file and go deep into the distribution model of some big malware campaigns. Last but not the least, go on deep learning of mismatch between Content-type and MINE Type, and do malware detection based on such mismatch.

ACKNOWLEDGMENTS

This research is supported by the National Science and Technology Support Program (No. 2012BAH46B02) and the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA06030200).

REFERENCES

[1] Google. Google safe browsing API, https://developers.google.com/safe-browsing/.

[2] C. Rossow, C. Dietrich, and H. Bos, Large-scale analysis of malware downloaders, Detection of Intrusions and Malware, and Vulnerability Assessment, 2013, pp. 42–61.

[3] Y. Ding, L. Guo, Ch. Zhang, Y. Zhang, H. Xue, T. Wei , Y. Zhou, X. Han, Poster: Classifying Downloaders, IEEE Symposium on Security and Privacy 2015.

[4] P. Vadrevu, B. Rahbarinia, R. Perdisci, K. Li, and M. Antonakakis, Measuring and Detecting Malware Downloads in Live Network Traffic, ESORICS'13, 2013, pp. 556–573.

[5] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S.-J. Lee, C. Kruegel, and G. Vigna, Nazca: Detecting Malware Distribution in Large-Scale Networks, Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS '14), Feb 2014.

[6] https://en.wikipedia.org/wiki/DOS_MZ_executable.

[7] VirusTotal, https://www.virustotal.com.