An Automated Framework for Command and Control Server Connection and Malicious Mail Detection

Lo-Yao Yeh
Department of Network and Security
National Center for High-Performance Computing
Taichung, Taiwan (ROC)
e-mail:lyyeh@narlabs.org.tw

Yi-Lang Tsai
Department of Director
National Center for High-Performance Computing
Tainan, Taiwan (ROC)
e-mail:yilang@narlabs.org.tw

***Abstract*—In recent Internet development, the amount of malware has increased significantly. There are more and more methods that hackers can use to infect personal computers to send spam mails, steal personal information, and launch Distributed Denial of Service (DDoS) attacks. This paper proposes a framework to strengthen security for users by integrating several online resources. The proposed framework can automatically prevent users from visiting malicious websites on the Internet Explorer browser. In addition, it can automatically detect the mail's source and attached files. Finally, if malware is connected to any Command and Control (C&C) servers, our framework is able to detect it by using an Application Programming Interface (API) hooking technique, and automatically kill it. By these methods, it will effectively restrain the scale of botnets and significantly reduce the risk of personal computers infection.**

***Keywords*—Network Security; Botnet; Email; API hooking;**

I. INTRODUCTION

As long as the number of computer devices increases, so does the number of compromised computers. If a device does not have any defense mechanisms, it will easy to become a bot. There are many ways for hackers to spread malware. For example, they can set up a malicious web site and use various means to induce individuals to browse it. Hackers can also attack normally benign websites. Most people worry less about benign websites, so they will generally trust all the files on the sites and perhaps download them. Alternatively, email is another popular way for hackers to launch attacks. Hackers can use email to spread links to users, luring them to phishing web sites, or masquerade as well-known companies to deliver emails to trick users and steal user account and password information. In addition, email attachments may also contain malicious files. When a user inadvertently executes the malware, their personal computer is turned into a bot and becomes part of a botnet. The bot behavior includes stealing personal information, sending spam and viruses, or launching Denial of Service (DoS) attacks. Therefore, the rapid expansion of botnets must be limited, and the user must have a security system to reduce their risks. This paper integrates the National Center for High-Performance Computing (NCHC) blacklist database [1], Virustotal [2], and the WHOIS server to reduce the probability of a user's computer becoming a bot. By parsing packets, our framework can automatically block a suspect page and warn users when they try to browse a known malicious webpage or download a malicious file. The analysis of web mail headers is used to determine whether or not mail is malicious, therefore protecting users. Finally, our system can monitor all applications on the PC to detect if any applications are trying to connect to a known Command and Control (C&C) server, which is used by a hacker to control the bots. Hence, we can effectively discover the potential malware and minimize the risk of infection.

II. PROPOSED SYSTEM ARCHITECTURE

Today, Microsoft Windows is the most popular operation system in the world, so our system architecture is designed for implementation on Windows XP and Windows 7. Our framework, as shown in Figure 1, is divided into two components, namely, the daemon program and the toolbar program. The daemon program monitors the network device to collect packets and uses the hooking technique to examine applications on user's computer. The toolbar is responsible to get the page information from the Internet Explorer (IE) browser.

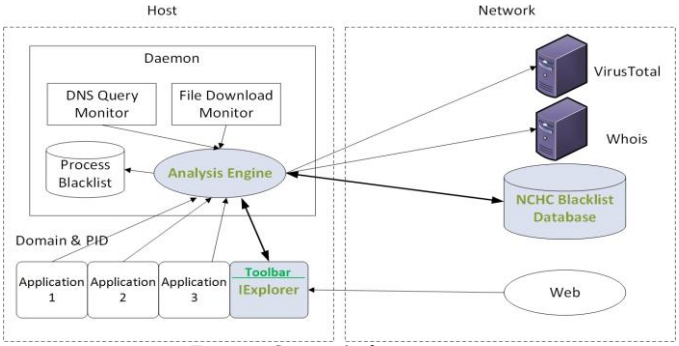

Figure 1. System Architecture

A. Analysis Engine

Our system integrates three different resources to complement our framework, including the online resource VirusTotal, the NCHC blacklist database, and the WHOIS server. Our framework uses the NCHC blacklist database and VirusTotal to check whether or not the domain or the files are malicious. In addition, our framework also takes advantage of WHOIS servers to detect forged mails.

B. Monitor Processes

Usually, malware will resolve Domain Name System (DNS names to locate the C&C server [4]. In our previous version of such a framework [5], our system parsed packets to check collected domains with the NCHC blacklist database. If a process connected to a known malicious domain, our system notified the user immediately. However, the drawback of our previous system is that we were unable to identify which process launched the connection. To solve this problem, we use

 ISBN: 978-1-61208-404-6

EasyHook [3] to integrate our system and hook each application. The daemon program can obtain domain names and Process IDentifier (PID) when the hooked process connects to any domain.

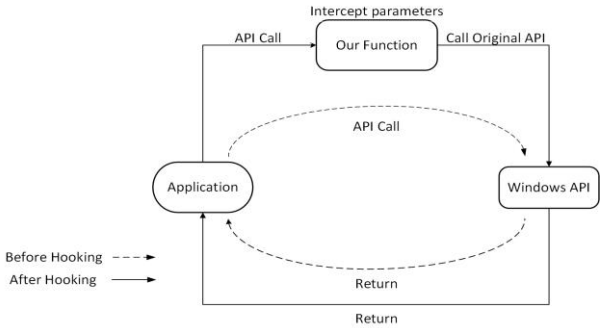

Figure 2. Execution flow with/without hooking.

When a process wants to resolve domain name service (DNS) names, it calls the specific API function to achieve this. Our system uses Easyhook to inject the pre-established data link layer or Dynamic Link Library (DLL) into every application's memory space. The DLL includes our function that intercepts parameters of the specific API function call, one of which is domain name. Figure 2 shows the execution flow of the function call before and after API hooking. The solid lines represent an execution flow without hooking. The dotted lines represent an execution flow with hooking. After the intercept parameters, we must to call original API to allow the application to finish its work. If any process connects to a C&C server, our system can obtain the malicious domain and PID from the process to show which specific process tried to connect to the malicious domain. The user can determine which one may be the malware and decide to kill this or not.

C. *Web detection*

- Blocking Malicious Pages

Our system can notify users when they are visiting malicious pages. Furthermore, the system can also actively block the malicious page before users visit it. The daemon commands every toolbar to monitor its own page. When the page wants to connect to a malicious web site, the toolbar blocks the page and asks users if they really want to visit the known malicious web site. If the answer is negative, the toolbar redirects the page to a blank page.

- Web Mail Detection

Email is one of the popular methods for hackers to attack computers. To protect users, our system has the following functions to detect malicious emails, and can be implemented on a web mail service.

1) Email Authentication

For mail reliability, most well-known companies, like Gmail and Yahoo! mail, use SPF [6] and DKIM [7] for mail authentication and spam filters. When the mail server receives mail, it will validate the identity of the sender, and then add an *authentication-results* header to the mail header. Our system can then check the mail header to examine results of SPF and DKIM authentication. In other words, when users receive mails from a well-known company, but the mail does not pass this authentication, users can be warned that the mail may be forged.

2) Mail Attachment

In order to prevent users from downloading malware from email, our system also examines the attachments. If a mail has attachments, it represents the sender requesting the content to be saved as a file. We can obtain the original files by decoding the mail body. Finally, our system will automatically upload the attached files to VirusTotal and the NCHC server to check whether the received files are malicious.

3) Received header

The *received* header is the most important header for tracking mail. When the mail server receives the message, the server adds a *received* header to the top of the header, recording the sender's name and IP address. However, because the sender's name can be manipulated, our system reads the *received* headers from top to bottom to detect whether the mail is forged. First, we check the sender's IP address item in each *received* header by WHOIS server to ensure that its domain name and sender's name are the same. Second, our system confirms whether they are malicious domains in the NCHC blacklist database. However, because some mail servers do not use their domain as the server name, we provide two further pieces of information, the *Alexa Rank* and *Page Views per Visit*, as a simple way for users to check the validity of the domain. In general, malicious domains unlikely have a high Alexa ranking and low page view value.

V. CONCLUSION

When using our framework, if a user connects to a known malicious domain, an infection alarm is issued to warn the user about the potential threat. Moreover, if malware already installed in the computer connects to the domain of a C&C server, our system can also find and kill the process. For email protection, our system not only detects the mail's source, but automatically scans attached files. As a result, we can effectively restrain the influence of botnets and reduce the chances of a PC becoming infected.

REFERENCE

[1] NCHC malware knowledge base. [Online]. Available from: http://owl.nchc.org.tw, 2015.03.25.

[2] VirusTotal. [Online]. Available from: https://www.virustotal.com/en/, 2015.03.25.

[3] EasyHook. [Online].Available from:http://easyhook.codeplex.com/, 2015.03.25.

[4] S. Shin, Z. Xu, and G. Gu, "Effort: Efficient and effective bot malware detection," Proc. IEEE INFOCOM, 2012, pp. 2846–2850.

[5] L.-Y. Yeh, Y.-L. Tsai, B.-Y. Lee, and J.-G. Chang "An Automatic Botnet Detection and Notification System in Taiwan", Intnational Conf. Security and Management, 2013, pp. 469-471.

[6] RFC 6652, Sender Policy Framework (SPF) Authentication Failure Reporting Using the Abuse Reporting Format, Proposed Standard, 2012.

[7] RFC 6376, DomainKeys Identified Mail (DKIM) Signatures Draft Standard, 2011.

 ISBN: 978-1-61208-404-6