Detecting Backdoors in Windows Processes

Mridul Ahuja
Jaypee Institute of Information Technology
Noida
mridul.ahuja@gmail.com

Anuradha Gupta
Jaypee Institute of Information Technology
Noida
anuradha.gupta@jiit.ac.in

ABSTRACT

In today's world, backdoors such as Trojan horses have become a common problem in computers, providing hackers unauthorized access to the compromised user's systems. Many of these viruses, in order to evade detection, inject themselves into legitimate processes. To detect these backdoors, various techniques such as signature-based detection, using sandbox, monitoring system calls and network traffic are used. In this work, we have implemented signature based detection technique to identify the backdoors in running processes on Windows operating system. For signatures, we calculated MD5 hash of the infected modules in the backdoors and stored them into a database. To identify whether process contains a backdoor, we compared the hash of the modules used by each running process with these signatures stored in the database. We also searched for ports being used by the suspected process to make sure if it actually contains the backdoor. We have tested the methodology on all 70 processes running on the system.

General Terms

Experimentation, Security

Keywords

Malware; Backdoor; Process; MD5 Hash; Signature.

1. INTRODUCTION

Computer science has undergone a tremendous improvement in the last decade, leading to a requirement for stringent security systems due to continuous invasion of new malwares. With the ever increasing security discipline, the viruses too have evolved enormously rendering the lower generation antivirus useless.

The early viruses were simple self-replicating programs that could spread through removable media [1]. With internet becoming a common medium of communication, a new form of malware evolved – Trojan, opening a new era for computer viruses. These Trojans, unlike viruses are non replicating in nature. They usually plant backdoors in systems, giving the attacker unauthorized access to the machine. Once a system is compromised, the attacker can spy on users, manage files, install additional software or dangerous threats, control the entire system including any present applications or hardware devices, shutdown or reboot a computer or attack other systems [2].

In this digital world, Security is a perpetually progressing subject where the hackers and antivirus companies are constantly trying to overcome each other. Backdoors and Trojans, unlike other viruses, hand over control of user's system to the attacker. There can be no better solution than to restrict their access and thereby protecting the data.

In this project, we will detect most commonly available Remote Administration backdoors.[1] These can be detected using signature-based detection [3], monitoring programs for suspicious behavior, dynamic analysis using FSM [4], using sandbox to emulate files in a controlled environment [5], and monitoring network traffic [6].

We have used signature based detection technique to identify backdoors. We will find the MD5 hash[2] of infected modules and perform the string matching operation with the hash of the modules of other processes. Also we are checking if the processes are using any ports to make sure they are backdoors.

2. METHODOLOGY

2.1 Signature Based Detection

Antivirus firms, on detecting a new backdoor, analyze it and extract a new signature of the file to their database. They compare each file on the system with each of the signatures collected. If a match is found, the antivirus is able to identify which malware it actually is. This technique is usually very effective for detecting the known backdoors [7].

We are also using the signature based detection approach, however, our methodology is slightly different from the traditional method. Our signatures are based on the modules used by the backdoors, rather than the backdoors themselves.

We gathered six popular backdoors that have been used for illegal purposes for the analysis, namely DarkComet Legacy, xRAT, NetOris, Apocalypse, Blackness and Blackshades [8, 9, 10, 11, 12, 13]. These programs were run on the system and their library files were examined.

These files were used as reference signatures. Table 1 shows the Trojans along with their signatures. These backdoors were then injected [14] into 17 legitimate processes. We then extracted the modules being used by each of the running processes and compared them to the signatures we collected. If a match was found, the process may be infected. Further to be sure, we checked whether these processes were listening to any open ports or not. If a port was found, the process was reported as backdoor. If no port was found, the process was suspected as a backdoor and

[1] Remote administration backdoors are softwares that provide the attackers complete control over the victims' system as if they have physical access to that system. The attackers have access to the registry and all files on the remote system

[2] MD5 is a widely used cryptographic hash function

requires more analysis. Flowchart of this method is shown in Figure 1.

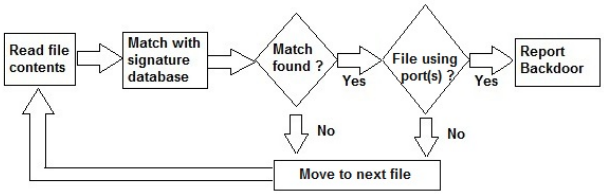

Figure 1 : Signature detection with port checking

Table 1 : Trojans with MD5 hash of infected module

Trojan	MD5 Hash
DarkComet Legacy	349E96A28B457ACF981EDA1C1F78AF26
Apocalypse RAT	97C91CD4180B82D66BFB70AC5027B6C4
NetOris	1BFEE5805CD7031CC58ED0B2BA5059D0
xRAT	110EE635588E8E75995505F10070C126
Blackness RAT	NOT FOUND
Blackshades	NOT FOUND

3. RESULT

The modules of Blackness and Blackshades could not be extracted as these programs registered themselves as system services. Signatures of all other backdoors were stored into our database. While reading running processes, System Idle Process, System, audiodg.exe, and BtvStack.exe being system processes were inaccessible. We have successfully infected 17 processes one by one with each of the available backdoors. These processes included chrome.exe, notepad.exe, acrobat.exe, iexplore.exe and winlogon.exe. The detection was successful for all backdoors except for 'Blackness' and 'Blackshades'. Table 2 shows the ports used by the infected processes.

The future work for this project is to identify the modules being used by Windows services to check if they are infected. Also we will try to detect backdoors on the basis of system calls being made by the programs.

Table 2 : Trojans and the ports they were listening to

Trojan	Default port
DarkComet Legacy	1604
Apocalypse RAT	1453
NetOris	6747
xRAT	1234
Blackness RAT	--
Blackshades	--

4. REFERENCES

[1] https://antivirus.comodo.com/blog/computer-safety/short-history-computer-viruses/

[2] http://www.2-spyware.com/backdoors-removal

[3] J.Nandhini, Dr.M. Nithya, Dr.S.Prabhakaran, Advance virus detection using combined techniques of pattern matching and dynamic instruction sequences, International Journal of Communication and Computer Technologies , 2013

[4] Gardåsen, Kjetil Tangen , Detecting RATs through dynamic analysis using Finite-State Machines, Gjøvik University College, 2014

[5] Martina Lindorfer, Clemens Kolbitsch, and Paolo Milani Comparetti, Detecting Environment-Sensitive Malware, Secure Systems Lab, Vienna University of Technology, 2011

[6] Nart Villeneuve, James Bennett, Detecting APT Activity with Network Traffic Analysis, Trend Micro Incorporated, 2012

[7] http://en.wikipedia.org/wiki/Antivirus_software#Signature-based_detection

[8] http://darkcomet-rat-legacy.software.informer.com/

[9] https://github.com/MaxXor/xRAT/releases

[10] http://sourceforge.net/projects/netoris/

[11] http://codhacker.blogspot.in/2013/12/apocalypse-rat.html

[12] https://cryptosuite.org/forum/visual-basic/41335-blackness-remote-administration-tools-v1-6-vb-net-source.html

[13] http://en.wikipedia.org/wiki/Blackshades

[14] http://resources.infosecinstitute.com/code-injection-techniques/