Low-cost detection of backdoor malware

Conference Paper · December 2017

DOI: 10.23919/ICITST.2017.8356377

Aspen Olmsted
Fisher College

Low-cost Detection of Backdoor Malware

Huicong Loi
Tandon School of Engineering
New York University
Brooklyn, NY 11220
hl2691@nyu.edu

Aspen Olmsted
Department of Computer Science
College of Charleston
Charleston, SC 29401
olmsteda@cofc.edu

Abstract—**Backdoor malware are programs that enable hackers to access unauthorized computer systems by introducing a backdoor. These hackers will use this access to steal company information for personal gain. This malware uses a variety of techniques to hide their presence, and computer security researchers use a growing number of exotic techniques to detect them. However, it is not necessary to expend valuable IT resources on expensive security solutions as most of these backdoors can be detected by simple checks. We tested a wide array of in-the-wild malware to verify the effectiveness of these checks.**

Keywords-component; backdoor; malware; Trojan; virus; detection; cheap; efficient; low-cost; simple; Windows

I. Introduction

Cybercrime is becoming increasingly sophisticated, graduating from the "smash-and-grab" attacks of yesteryear. One of the most dangerous attacks detailed by Ping et al. involves hackers installing backdoor malware on computers in targeted companies [1]. These backdoors use a variety of techniques to avoid detection [2] and allow the hackers free access to the victim's computers. Due to their stealthy nature, the backdoors allow the hackers to exfiltrate valuable business secrets while remaining undetected for a long time. To combat malware proliferation, computer security researchers have developed a diverse set of techniques to aid detection efforts [3].

In this paper, we develop an efficient method to detect stealthy malware, but without any reliance on exotic technologies or tools. This is accomplished primarily by targeting the command-and-control feature of backdoor malware. We then prove the technique by testing it out on a variety of malware samples captured from the wild.

The rest of the paper is organized as follows: Section II reviews related work. Section III describes the motivation for our work. Section IV describes the data collection mechanism and analysis. Finally, Section V will provide a conclusion and future work.

II. Related Works

Idika et al. [3] performed a review of commonly-used heuristic methods used to detect malware activity. In their review, they noted that each heuristic had inadequacies, of which a high false-positive ratio and a large startup cost were the most common issues. They also discussed methods that could be used to defeat the various heuristics. Some heuristics that were reviewed were: suspicious API call sequences, the frequency of common machine code sequences used in malware as well as program control-flow graphs.

Firdausi et al. [4] analyzed the effectiveness of using machine-learning to classify and detect malware activity. Their process consisted of sending both benign software as well as malware samples to a binary-analysis service which would derive various features and attributes. This information would be weighted and then used with various classifier techniques to predict if a particular program was a malware or not. The authors report a 95% success rate with the J48 classifier technique. However, it has a high running cost as each new item to be analyzed needs to be subjected to the binary-analysis service.

Christodorescu et al. [5] reviewed the effectiveness of some commercial antivirus products such as Sophos, Symantec, and McAfee. They found that commercial products were heavily dependent on signatures and simple obfuscation methods such as renaming variables and inserting garbage sequences would allow a previously flagged malware program to evade detection.

III. Motivating Example

Cybercrime is a very real problem in today's environment, but private-sector cybersecurity teams often operate with inadequate resources as security is often seen as a cost rather than a requirement [6]. Such teams might not be able to afford the newest cybersecurity solutions, which are becoming increasingly necessary due to the proliferation of malware obfuscation techniques [2] and the ineffectiveness of relying on antivirus software alone [5]. Having an efficient but low-cost method of detecting malware threats is of critical importance to these teams.

IV. Implementation

We rely on a different approach from the related works, as the solution is intended for teams with limited resources. Due to this constraint, our solution is not intended to be able to detect all types of malware nor to achieve 100% intrusion prevention. The solution is developed for the Windows platform as this is the most common desktop environment.

For most data theft scenarios, a common requirement is that some malware that facilitates an internet connection must be present on the victim's computer. This connection is used by the hackers to control these victim computers remotely and to run other hacking tools to perform other infiltration tasks [1]. The malware that creates the internet connection is commonly called a backdoor. By targeting these backdoors, we sever the hackers' access to the victim computers and ensure safety. It is not essential to be able to detect the other types of hacking tools as the hackers would not have any means of controlling them without the backdoors.

The easiest way of finding out if a backdoor is running on your system is to see if some sort of internet connection has been established. All backdoors, except the most advanced kernel rootkits, reveal themselves in this manner. As even expensive cybersecurity solutions have difficulty dealing with kernel rootkits [7], so we do not see this as a limitation of our solution. However, merely creating an internet connection is not sufficient to flag a process as malicious. Internet browsers and various updater systems have perfectly legitimate reasons for creating connections. We need to derive additional characteristics that can differentiate between legitimate and malicious processes.

One question to ask is, "Why not filter by the internet connection's destination"? It is not practical to create a list of all malware sites in the world as there are simply too many and the exact number is very fluid. Some malware camouflages their traffic by using legitimate internet services such as Twitter or Google Drive.

To derive these distinguishing characteristics, we obtained malware from 'theZoo' malware archive and randomly selected 20 for testing. We developed a minimal list that satisfied detection for all of the 20 malware that was tested.

1. Process must make an internet connection
2. Svchost.exe makes an internet connection, but its parent is not services.exe.
3. The process makes internet connection but either does not have an existing parent process, or its ancestor is not explorer.exe.
4. The process makes internet connection but is a common Windows process that should not have an internet connection (explorer.exe, dllhost.exe, regsvr32.exe).
5. Subject all other processes to VirusTotal checks.

We used free software tools to implement our solution. TCPLogView was used to log internet connections as it had a very low CPU load and could be kept on without affecting user experience. A WMIC command was used to check the parent of flagged processes. Finally, VirusTotal has Python API examples which can be used to automate the task of submitting files for evaluation. One area of inefficiency could be the requirement of a large number of files needing to be submitted to VirusTotal for checks. One mitigation could be to develop a database of hashes of files that have been submitted to VirusTotal. Each file would only need to be submitted once, and subsequent checks could be referred to from the database.

We tested the implemented solution against 10 randomly selected malware from 'theZoo' malware archive.

TABLE I. TEST RESULTS

Malware	Detected	Rule
BlackEnergy2.1	no	-
Poweliks	yes	3
Rustock	yes	5
Trojan.Asprox	yes	2
Trojan.Bladabindi	yes	3
Trojan.Kovter	yes	3
Trojan.Loadmoney	yes	5
Win32.Sality	yes	4
Win32.Zurgop	yes	2
Win32.Lephic	yes	4

We achieved a 90% success rate. The 'BlackEnergy2.1' malware circumvented our checks as it installed and ran as a legitimate Windows service.

V. CONCLUSION AND FUTURE WORK

In this paper, we proved that our solution is effective against a variety of backdoor malware, even though we only used simple tools and techniques. However, there are still improvements to be made to the system.

- Dealing with malware that is installed as services
- Dealing with malware that injects themselves into the memory of internet-capable processes

REFERENCES

[1] P. Chen, L. Desmet and C. Huygens, "A Study on Advanced Persistent Threats," in *IFIP International Conference on Communications and Multimedia Security*, Portugal, 2014.

[2] I. You and K. Yim, "Malware Obfuscation Techniques: A Brief Survey," in *2010 International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, Fukuoka, Japan, 2010.

[3] N. Idika and A. P. Mathur, "A survey of malware detection techniques," Purdue University, West Lafayette, Indiana, 2007.

[4] I. Firdausi, C. Lim and A. Erwin, "Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection," in *Second International Conference on Advances in Computing, Control, and Telecommunication Technologies*, Washington, DC, 2010.

[5] M. Christodorescu and S. Jha, "Testing Malware Detectors," in *ACM SIGSOFT International Symposium on Software*, Boston, 2004.

[6] A. Etzioni, "The Private Sector: A Reluctant Partner in Cybersecurity," in *Privacy in a Cyber Age*, vol. 28, New York, Palgrave Macmillan, 2015, pp. 58-62.

[7] B. Schneier, "The Failure of Anti-Virus Companies to Catch Military Malware," [Online]. Available: https://www.schneier.com/blog/archives/2012/06/the_failure_of_3.html. [Accessed 21 July 2017].