Evaluating Capabilities of Rootkits Tools.

Article · November 2016

CITATION
1

READS
944

2 authors:

John Alhassan
Federal University of Technology Minna
59 PUBLICATIONS **149** CITATIONS
SEE PROFILE

Sanjay Misra
Covenant University Ota Ogun State, Nigeria
583 PUBLICATIONS **3,740** CITATIONS
SEE PROFILE

Evaluating Capabilities Of Rootkits Tools

[1]J. K. Alhassan, [2]S. O. Subairu and [3]S. Misra

[1] Department of Cyber Security Science, Federal University of Technology, Minna, Nigeria
E-mail:jkalhassan@futminna.edu.ng
[2] Information Technology Services, Federal University of Technology, Minna, Nigeria
[3] Covenant University, Otta, Nigeria

Abstract: Rootkit is a fatal malware devouring user and kernel mode kind which inclines to take complete control of a compromised system by means of various infection and evasion techniques. Several detection algorithms has been offered and joined into the anti rootkit tools with many degree of performance in handling rootkit incidence. There is a severe rise in the rootkit attack with irregular rootkit samples such as, zeroaccess, darkmegi, tdl-4 and xpaj.mbr with each one having different impact on the internal structure of an operating system. Therefore, in this study analysis of rootkits tools were carried out using active detectors tools and malware forensic analysis tools, applying system scanning, network scanning and malware forensic analysis methodology. Altogether the samples rootkit have one or more rootkit detectors to handle their incidence though at a varied performance rate except darkmegi. Though two of the detectors were able to detect its presence on a compromised system, but failed in removal attempt.

Keywords: Rootkits, infection, detectors, detection, network scanning,

Introduction

Rootkits are a remarkably dangerous kind of malware owing to the fact that they might be intelligent to shield their existence on the host operating system. They can exploit stealth applied sciences, permit malicious regeneration through spyware and additional obvious types of malware unnoticed. The moment rootkit gained entree to a system, it can be very hard to trail and do away with them [1]. The word "rootkit" (a suite for gaining a "root", or administrator access to the goal system) initiated within the UNIX world, where "root" system access entails the finest likely level of procedure influence rights, to administrators, [1].

The second portion of the expression, "kit", expresses that groups of program examples exist that somebody can collect either free of charge or for an amount and adapt for use together with their own malware to cloak that software's activities. In numerous occurrences rootkits are circulated in an open-source, which means that even unprofessional programmers can effortlessly manipulate rootkit code; for illustration, to exclude discovery through anti-virus program that is surveillance for virus signatures, on the grounds that the rootkit would shield the virus's signature [2].

Nearly rootkits modify operating system (OS) application program interfaces (APIs) by indicating the address of these APIs to point to their infected code. This can also be finished equally in user mode (Applications Runs) and kernel mode (location for running device drivers) and is usually known as hooking. The instant an application lunched a hooked API, the OS look through the system service dispatch table (SSDT) in the kernel mode and the import address table (IAT) in the user mode for the address of the API. The code at that address is then executed. If a rootkit has hooked the API, its code is then runs, in its place of the expected functionality. This licences the rootkit to interrupt requests that might disclose its existence, [3].

Rootkit serves as the entryway to other fatal malware, its silent technology make it problematic for most antiviruses that mainly used signature-based detection algorithm to detect. This research is on comparative study analysis of Rootkit infection, detection and removal techniques and by so doing offer an appropriate method in handling Rootkit occurrence using the sample rootkits, [4].

Literature Review

[5] Pointed out that, integrity detection grants a replacement to equally signatures and heuristics. That it relies upon assessing a file system or memory with recognized, reliable baseline. The current and baseline snapshots liken and the variances are taken as proof of malicious action. However, the integrity checker lacks the capability to identify the source of the reasons that has caused the variations.

[6] stated that Copilot is hardware founded detection software which began at the University of Maryland and has bred an autonomous company. Currently, Copilot is within the form of a PCI card that is set up on the host been watched for rootkit movement. The reason of the PCI card is to continue as impartial of the possibly overthrown operational atmosphere. To attempt this, the PCI card have CPU of its own and makes use of Direct Memory Access (DMA) to probe the system watching for rootkit conducts such as hooks within the SSDT, changes to kernel services (using kernel reliability tests), and changes to crucial memory structures as the circumstance of a DKOM attack.

[7] disclosed that, the contemporary violence model of rootkit and other malware has developed to robust threat than before, that the malware writers has well-defined many ways to convey their malicious

[8] noted that malware such as rootkit and others adopted an complication method in order to hide their malicious code and avoid discovery by antimalware tools and this method differs according to the methods implemented but with one aim.

According to [9], dead code insertion occurs when a garbage code that are not active is added into the original code of the malware, to alter its presence but its malicious behavior is unharmed.

[10] is an independent organization that carried out performance testing of current antivirus software to see whether they fulfill the security protection they promise. The comparative analysis is conducted periodically and their reports are helpful in the ranking of antivirus. Their latest comparative test containing malware is inadequate and not intended for advance malware or administrator instead for beginner home user. Also, their test is restricted to antivirus capacity to sense malware.

[11] work is on comparative analysis of rootkit detection techniques. Five samples of rootkit were used in the research with about twenty rootkit detector, although most detectors used were not dynamically maintained; hence their detection competence could not be trusted upon. Ranking of the detectors were offered with those tools been sustained on top of the rank. Samples of rootkit used were actually threat at the time of the research but more dangerous samples are now in the wild such as zeroaccess , TDL-4, darkmegic, xpaj-mbr and host of others. .

Methodology

The materials used for this research are three computers systems, one switch and Four Rootkits sample namely; ZeroAccess, TDL-4, Xpaj-MBR and Darkmegi. Fifteen Rootkit detectors were used, namely: aswMBR, TDSS killer, Gmer, Rootkit remover, Bootkit remover, Malwarebytes Anti-rootkit, Comodo cleaning essential (cce), AVZ4, Vba32 Antirootkit, Emco Malware Destroyer, Stinger, Roguekiller, Unhackme, Regrun Plantinum, Rising Antivirus, Malware Forensic tools, Microsoft Kernel Debugger (KD.Exe), cRegistry comparison and Perfmon. Nmap / Zenmap and Netsat were the network scanning tools used, while DBAN and Diskwipe were the Hard Disk Drive wiping tools used.

In scanning the system first, rootkit was installed on a clean system, thereafter verification was carried out to determine fruitful installation by performing a kernel mode debugging session. Afterward rootkit installation was established, apiece of the anti-rootkit tools were install and scanning of the infected system took place. Once a tool detects a root kit, such tool is detected whether it has elimination codes. Furthermost frequently through the internet, via social networks like Facebook and others, through open source download, freeware and social engineering.

technique, if it has, then removal of such rootkit is attempted. To check the fruitful removal, the tool is used to scan the system drive once more. If no rootkit is found, the drive is then rubbed using HDD wiping tools such as DBAN. Then operating system is re-installed, followed by infecting it with rootkit and scanning resume using other detectors.

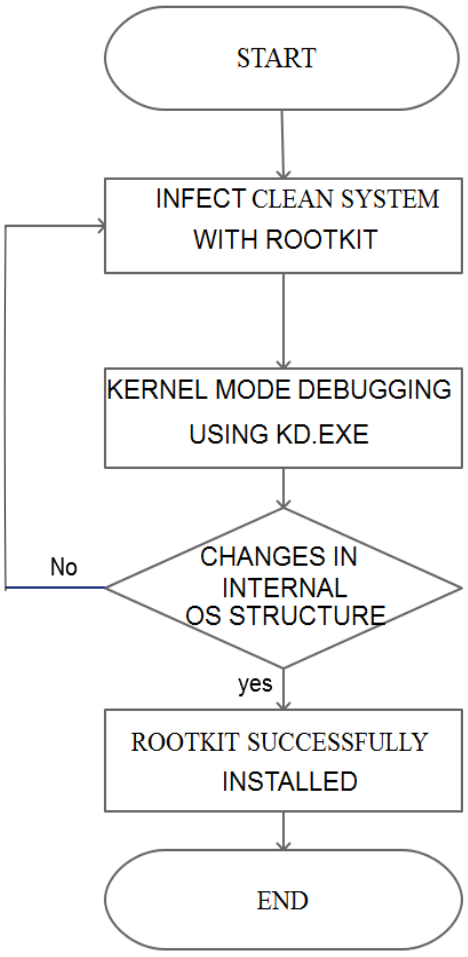

Figure 1: Forensic Analysis Flowchart

Two systems were used in carrying out this experiment, one of the systems was infected with one of the test samples rootkit and the other system (CS) as a clean system. Netstat was used as the network software on the infected machine, while Nmap and Zenmap were used as the port scanning software on the external remote system. The Network scanning was as follows:

1. IP address on the infected system was identified by using ipconfig on command line.
2. Then a command line was opened and executed: netstat -a -n >> [name of test sample Rootkit] _netstat.txt, this was an output to a log file.

3. Zenmap software GUI version of Nmap was opened and the IP address of infected system as obtained in 1 was inserted, intense scan option was selected and then scanning commenced.

Results And Discussion

Table 1 displays the scan result of a clean uninfected system by the numerous rootkit detectors; scan time and false positive were observed.

Table 1: Scan outcome for uninfected system

S.N	Detectors	Version	Scan time (HH:MM:SS)	False positive
1	Avz antiviral Tootkit	4.43	00:00:32	No
2	Comodo cleaning essentials	2.5.242177.201	00:14:46	No
3	Emco malware destroyer	7.5.15.1950	00:00:23	No
4	Vba32arkit	3.12.4.0	00:00:23	No
5	aswMBR	1.0.1.2290	00:00:29	No
6	Gmer	2.1.19357	00:08:16	No
7	Malwarebytes	1.09.1.1004	00:04:30	No
8	Mcafe rootkit removal	0.8.9.174	00:00:10	No
9	Bootkit Removal Tool	3.0.2.2.011	00:00:05	No
10	Kaspersky Tdsskiller	3.0.0.44	00:00:39	No
11	Unhackme	7.71	00:00:12	No
12	MacAfee stinger	12.1.0.1534	00:01:21	No
13	Roguekiller	10.6.5.0	00:02:15	Yes
14	Regrun Platinum	7.7	00:00:10	No
15	Rising Antivirus	23.01.24.80	00:01:01	No

From the outcome in table 1, all the detectors do not illustrate false positive except roguekiller which recognize some windows file as a threat. This further put following scan outcome of roguekiller into further confirmation to essentially check if the file or files identify are actually malicious or not. Similarly the result displays bootkit removal tool to have the smallest scan time, perhaps as result of its detection algorithm which is limited to signature base.

The comparative analysis model was used for rootkit detectors ranking and the results presented base on the general performance on the samples rootkit. As shown in table 2, simple mark is allotted to each parameter of ranking, such that two points to detection, two points to rootkit successful removal and one point is subtracted from any rootkit that reported false positive.

Table 2: Rootkit detectors ranking

S. N	Detectors	Detection	Removal Successful	False Positive	Total Score
1	Avz antiviral Tootkit	0	0	0	0
2	Comodo cleaning essentials	6	2	0	8
3	Emco malware destroyer	2	2	0	4
4	Vba32arkit	0	0	0	0
5	aswMBR	2	0	0	2
6	Gmer	8	0	0	8
7	Malwarebytes	4	4	0	8
8	Mcafe rootkit removal	0	0	0	0
9	Bootkit Removal Tool	2	0	0	2
10	Kaspersky Tdsskiller	2	2	0	4
11	Unhackme	2	2	0	4
12	MacAfee stinger	6	6	0	12
13	Roguekiller	4	2	-1	5
14	Regrun Platinum	2	2	0	4
15	Rising Antivirus	2	0	0	2

As shown in table 2, the best performed rootkit detector on the samples rootkit is the mcAfee stinger, apart from been dynamically reinforced, it similarly uses most of the known detection algorithm. It discovery level is quicker liken to other detectors used in this research work. McAfee extremely enhanced on this tool as liken

to rootkit removal which really perfomed awfully poor against the four examples rootkit. Three additional detectors performance were valued very good as shown in table 2. They are comodo cleaning vital, gmer and malwarebytes. Their detection algorithm required to be advance for enhanced detection performance, also detection rate of gmer and comodo cleaning essentially needed an improvement to increase efficiency. Roguekiller, is rated well, however it is the only detector amongst other detectors that were used for the research to have reported false positive. Thus, its detection algorithm should be fine-tuned to remove this hindrance. Emco malware destroyer, kaspersky tdsskiller, unhackme and regrun plantinum were all rated as fair in their performance. This little performance is ascribed to implementation of their detection algorithm within the tool, particularly for emco malware destroyer and kaspersky tdsskiller which both uses all the known detection algorithm, as that of mcAfee stinger, yet their performance fall below that of stinger. Bootkit removal, aswMBR and rising antivirus were rated poor with total score of two points. Though their scan time is good but development needed on their detection mechanism considering the trend of modern rootkit. Avz, vba32ark and mcAfee rootkit removal shows the worst performance against the four samples rootkit. Vba32 performance is not unexpected as tool is no longer been vigorously reinforced, also earlier researcher point out in their findings that both vba32 and avz shows 30xtremely poor performance against other types of rootkit samples. McAfee has improve on rootkit removal with the introduction of stinger which look very promising in the battle touching rootkit and other kinds of malware.

Conclusion

All the samples rootkit have one or more rootkit detectors to handle their incidence though at a varied performance rate as shown in table 2, except darkmegi. Though two of the detectors were able to detect its presence on a compromised system, but failed in removal attempt. This is a proof of the characteristics of darkmegi with a strong technique to deny delete access to any of its files and registry keys it created on a compromised system. Therefore, more effort is needed in removal algorithm to get rid of this kernel rootkit.

References

[1] Chris, R. (2006, May 22). Inside Windows Rootkits. Vigilantmind Inc. Retrieved from http://repo.hackerzvoice.net/depot_madchat/vxdevl/library/Inside%20Windows%20Rootkits.pdf, 1-18.

[2] Ashwin, R. (2008, September 2). Detecting kernel rootkits. Master's Thesis Proposal Dartmouth Computer Science Technical Report TR2008-627. Retrieved from http://www.ists.dartmouth.edu/library/409.pdf, 2-5.

[3] Uppal, D., Mehra, V., Verma, V. (2014). Basic survey on Malware Analysis,Tools and Techniques. International Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1, 103-111.

[4] Kirti, M, Saroj, H. (2013). A Survey on Techniques in Detection and Analyzing Malware Executables. International Journal of Advanced Research in Computer Science, SoftwareEngineering. Volume 3, Issue 4 ISSN: 2277 128X, 422-428.

[5] James, B., Sherri, S.(2010). Windows Rootkits. Retrieved from http://www.symantec.comconnect/articles/windows-rootkits-2005-part-two

[6] Hejazi, S. (2009). Analysis of Windows memory for forensic investigations (Master's thesis).Retrieved from http://spectrum.library.concordia.ca/976393/1/MR63196.pdf, 1-100.

[7] Rehman, R., Hazarika, D., Chetia, G. (2011). Malware Threats and Mitigation Strategies: A Survey. Journal of Theoretical and Applied Information Technology. Vol. 29 No.2 ISSN: 1992-8645, 69-72

[8] Marpaung, J.A.P., Sain, M., Hoon-Jae, L. (2012). Survey on Malware Evasion Techniques: State of the Art and Challenges. Advanced Communication Technology (ICACT), 2012 14th International Conference. ISSN: 1738-9445, 744-749

[9] You, I., Yim, K. (2010). Malware Obfuscation Techniques: A Brief Survey. International Conference on Broadband, Wireless Computing, Communication and Applications, 297-300.

[10] AV-Comparative. (2014). Retrieved from http://www.av-comparatives.org/wp-content/uploads/2015/01/avc_sum_201412_en.pdf

[11] Arnold, T. M. (2011). A Comparative Analysis of Rootkit Detection Techniques (Master's thesis). Retrieved from http://sce.uhcl.edu/yang/research/A%20Comparitive%20Analysis%20of%20Rootkit%20Detection%20Techniques.pdf