Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN : 2348-4470
p-ISSN : 2348-6406

International Journal of Advance Engineering and Research Development

Special Issue on Recent Trends in Data Engineering

Volume 4, Special Issue 5, Dec.-2017

Novel Implementation of Hybrid Rootkit.

Prof Dr.Kishor Wagh[1], Shashank Patwardhan[2], Siddesh Garsund [3], Shivani Bhakare[4] and Shravani Jadhav[5]

Computer Department, AISSMS IOIT

Abstract — *Statistics show that however malware detection are detecting and preventing malware, they do not guarantee of a 100% efficiency in thedetection and / or prevention of malware. This is especially the case when it comes to rootkits that can manipulate the operating system such that it can distribute other malware, hide existing malware, steal info, hide itself, disable anti-malware software etc. all without the knowledge of the user. This paper will demonstrate that by implementing hybrid rootkits or any other type of malware, a researcher will be able to better understand the techniques and vulnerabilities used by an attacker. Such information could then be useful while implementing anti-malware techniques.*

Keywords-*Hybrid rootkit, malware, stealth, security, Trojans, Amazon Web Services.*

I. INTRODUCTION

A Rootkit[1] is a suite or collection of one or more programs that allows a third party to hide files and activities from the administrator of a computer system. An intruder takes advantage of one or more known vulnerabilities on a particular computing platform to deliver and install the rootkit. Once the rootkit is in place, the intruder can use the infected system while remaining undetected.

A battle has started between the defenders and attackers of computer systems. The term attacker refers to one who tries to compromise a system and intends to carry out malicious activities on that system that remain invisible to defenders. At the same time, defenders search for successful attacker by searching for signs of system compromise or malicious activities.[6]

In this paper, it is presumed that the perspective or intention of the attacker isto try to runamalicious software and avoid the detection of the same. By assuming this intention of the attacker, we try to make the defenders understand the threat possessed by a new class of rootkits, so that they can protect their system.

II. RELATED WORK

Yong Wang, et al [2] proposed a Virus Analysis on IDT-Hooks of Rootkits Trojan Interrupt Descriptor Table (IDT) hook which isa rootkit technology at the kernel level of Trojan. They have compared IDT structure and programs to understand how Trojan interrupt handler code can respond the interrupt vector request. Finally, they analysed the IDT-hook detection methods of rootkits Trojan by Windbg or by other professional tools. IDT hooking and unhooking is explained deeply in the proposed structure which helped in detection of rootkit Trojan IDT.

Paul Watters, et al[3] proposed aWindows Rootkits: Attacks and Countermeasures. In this paper, Windows XP is one of thepopular operating system in the world today and hence rootkits have been a major concern for XP users. We have identified some of weaknesses in windows XP architecture that rootkits exploit and then evaluate some of the anti-rootkit security features that Microsoft has unveiled in vista and 7. The proposed structure has mentioned some of the anti-rootkit features that Microsoft has introduced in the last few years, but there are some architectural, and software defects that rootkits could take advantage of, so Microsoft needs to work on that.

Sebastiaan Von Solms, et al[4] proposed an Implementing Rootkits to Address OS Vulnerabilities. This paper demonstrates the steps required to in order to develop two rootkits. One rootkit can sabotage a Windows OS and the other can disable anti-malware programs and log the keys pressed by the programs. The steps demonstrated the vulnerabilities the rootkits exploited namely the kernel, sharing memory, the registry, the Windows boot loader, system messages and the user. This information could then be useful for researcher to understand working of rootkits and for implementing anti-malware techniques. Proposed structure gives a good way of implementing rootkit without being detected by most of the rootkit detection systems by loading virtual machine monitor before the operating system.

Shawn Embleton, et al[5] proposed a SMM Rootkits: It is a newly discovered breed of OS Independent Malware. In this paper, a proof of concept SMM (System Management Mode) rootkit is developed and presented. This proof is limited to only those systems which have single processor and works on only PS/2 keyboards. It helps the security researchers to better understand the depth and scope of problems posed by emerging class of OS independent malware. The proposed

structure has stated that the discovered SMM rootkit can be presented as a new breed of OS independent malware. Thus it may give an valuable contribution to an effective an efficient multi-vector rootkit attack which has capability to target a huge psrt of current systems in the market.

Samuel T. King, et al[6] proposed that implementing a new type of malware that gains qualitatively more control over a system, called as virtual-machine based rootkit (VMBR) that installs a virtual machine monitor below an existing operating system and loads the original operating system into a virtual machine. To run the VMBR it should be loaded before the loading of the target operating system. After VMBR is loaded, it boots the target operating system using the VMM without the concern of the victim. As a result, the target runs normally, but the VMBR sits silently beneath it. The proposed structure has a good method of implementing rootkits which is not detected by many rootkit detection systems and also the it has a greater control over the victims computer.

YoheiAkao, et al[7] proposed a Kernel Rootkits Detection Method by Monitoring Branches Using Hardware Features. This paper proposes a new method to detect kernel space using hardware features of commodity processors. This approach helps in detecting kernel rootkits immediately and also reduces the damages to a minimum. This method uses the Last Branch Record of Intel processors for monitoring the branch record in kernel space. In future work, it will handle case in which interrupts occur and also evaluate the performance of this method. Implementing new methods for detecting kernel rootkits have been proposed in this structure which minimizes the damages, therefore this paper stands out differently.

Vinod Ganapathy, et al[8] proposed Detecting Kernel-Level Rootkits Using Data Structure Invariants. This experiment demonstrates the feasibility of auto-mated generation of integrity specifications for the kernel data structures. The invariants inferred by this approach can serve as the starting point for the kernel experts, who can further refine these specifications. We have also found that the automatically generated invariants were quite precise. Other techniques for rootkit detection have focused on detecting control data modifications and, therefore, failed to detect some rootkits whereas the proposed structure stands differently by presenting a technique to detect a variety of rootkits that modify both control and non control data.

Christian Platzar, et al[9] proposed a Practical Approach for Generic Bootkit Detection and Prevention. In this paper, a novel approach to detect and prevent bootkit attacks during the infection phase is presented. It presents results of a preliminary evaluation on this approach using a Windows system and the leaked Carberpbootkit. The approach is based on VMI for boot process emulation and monitoring to detectbootkit infections. The system consists of mainly two major components: an on-write disk access driver, protecting disk sectors that are responsible for booting and the detection engine that evaluates whether the system is infected or not. This approach is yet not completed and is looking forward to implement it in a commercial scan engine. The proposed structure has stated that the SMM rootkit can be viewed as a new breed of OS independent malware related to VMBR and BIOS rootkits.Thus it may contribute to an effective multi-vector rootkit attack which is capable of targeting a large subset of current systems in the market.

Types of Rootkits:

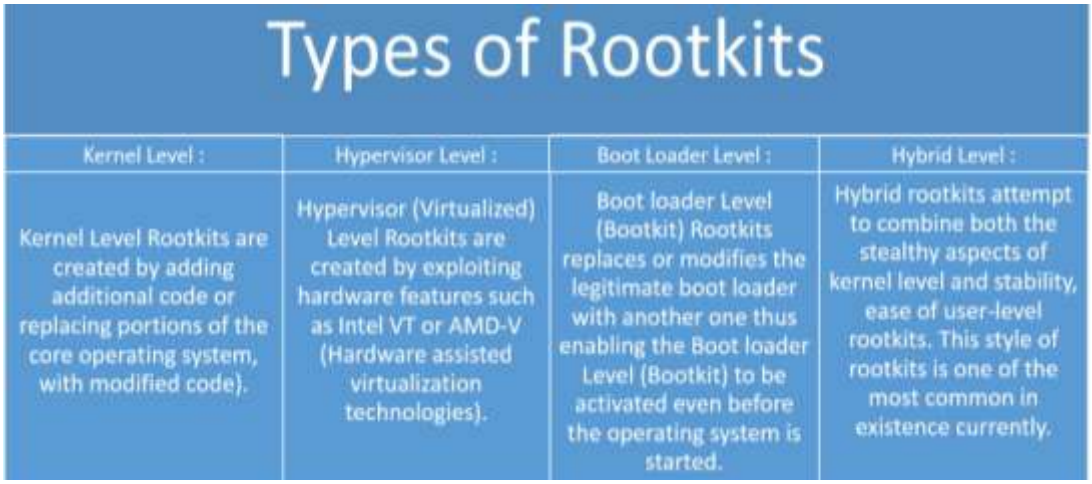

Fig 2. Classification of rootkits

Form above related work it seems that none of the above authors have performed hybrid rootkit implementation and hence this paper will stand out from others.

III. PROPOSED SYSTEM ARCHITECTURE:

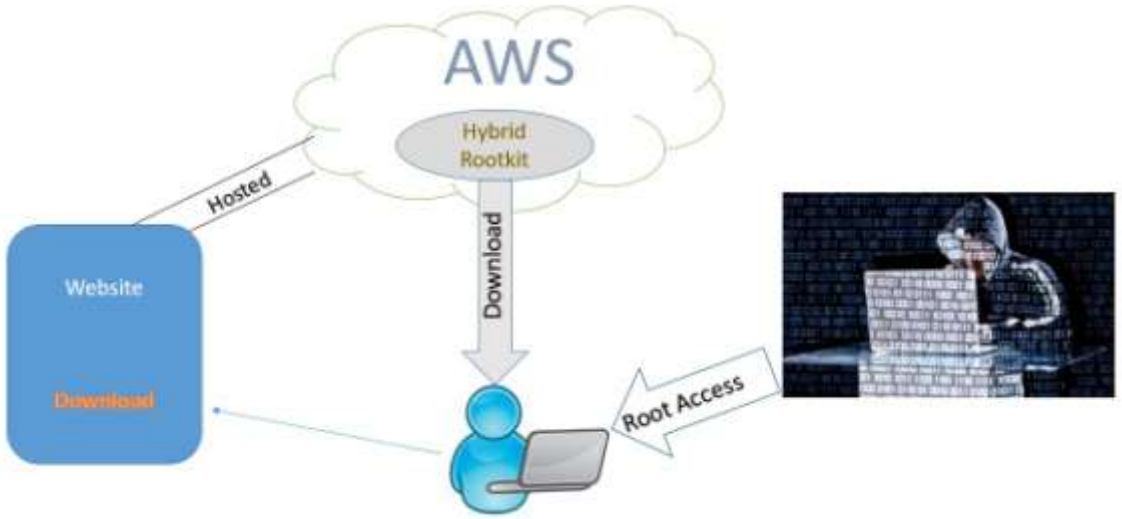

Fig 3.1. An overview figure of the proposed architecture

The figure shows an overview of the proposed architecture which consists of mainly four components.

The components are:

1. Attacker: The user who will gain the root level access of victims system.
2. Victim: The user whose system will be attacked to gain his/her root level access.
3. Website: A static website that is hosted by the attacker to phish the victim for the purpose of gaining root level access of victims system.
4. AWS (Amazon Web Services): The cloud provider that is used to host the website and also store the rootkit.

Methodology:

1. At first, we develop a Hybrid Rootkit. To develop a hybrid rootkit we have to develop a combination of the above mentioned rootkits (i.e. kernel, hypervisor, bootkit)
2. After we have developed a Hybrid Rootkit, we have to patch it to a legitimate file (Here legitimate file is any file like any executable file).
3. Deployment phase:
    i. Deployment of the rootkit on a website that has clickjacking enabled.
    ii. Amazon Web Storage is used for storage and hosting purposes.
4. Now, when the victim will fetch the hosted website, wherever he/she clicks, the legitimate file he/she is looking for will be downloaded with rootkit patched to it.
5. When the user executes this legitimate file, a connection request will be received at the attackers end and we can establish a connection to the victim's machine having root access in stealth mode.
6. The attacker can hence perform his required operations such as obtaining log files, passwords or any other confidential information on the victims system without the knowledge of the victim.

IV. CONCLUSION:

In this paper, we have developed, patched, deployed and demonstrated hybrid rootkit implementation for multiple operating systems (Windows, Fedora, MacOS/Android) using cloud platform. This product shall assist Federal Agencies for investigative purposes. It may also give an advantage of remaining anonymous and will provide a greater control of target system as we are implementing hybrid rootkit. Researchers will get detailed knowledge about rootkits and their working methodologies which will intern help them for creating a watertight anti-malware or rootkit detection system.

REFERENCES

[1] Computer Security and Rootkits: JameelAlsalam, Somnath Banerjee, Grant Musick and RaresSaftoiu

[2] Virus Analysis on IDT Hooks of Rootkits Trojan: Yong Wang, DawuGu, Wei Li, Jing Li, Mi Wen. 978-0-7695-3686-6/09/ 2009 IEEE. DOI 10.1109.

[3] Implementing Rootkits to Address OS Vulnerabilities: ManuelCorregedor and Sebastiaan Von Solms. 978-1-4577—1483-2/11/ 2011 IEEE.

[4] Windows Rootkits: Attacks and Countermeasures**:** Desmond Lobo, Paul Watters, Xin-Wen Wu, Li Sun. 978-0-7695-4186-0/10/ 2010 IEEE. DOI 10.1109

[5] SMM Rootkits: A New Breed of OS Independent Malware: Shawn Embleton, Sherri Sparks, Cliff Zou. NSF Cyber Trust Grant CNS-0627318 & Intel Research Fund.

[6] SubVirt: Implementing malware with virtual machines: Samuel T. King, Peter M. Chen, Yi-Min Wang, Helen J. Wang.

[7] Proposal of Kernel Rootkits Detection Method by Monitoring Branches Using Hardware Features: YoheiAkao, Toshihiro Yamauchi.

[8] Detecting Kernel-Level Rootkits Using Data Structure Invariants: AratiBaliga, Vinod Ganapathy, and LiviuIftode.

[9] A Practical Approach for Generic Bootkit Detection and Prevention: Bernhard Grill, Christian Platzar and JurganEckel.

