Poster: CloudSkulk: Design of a Nested Virtual Machine-based RITM Attack

Joseph Connelly
Boise State University
joeyconnelly@u.boisestate.edu

Haining Wang
University of Delaware
hnw@udel.edu

Jidong Xiao
Boise State University
jidongxiao@boisestate.edu

***Abstract*—When attackers have compromised a system and have some certain control over the victim system, retaining that control and avoiding detection becomes their top priority. To achieve this goal, various rootkits have been proposed. However, existing rootkits are still detectable as long as defenders can gain control at a lower-level, such as the operating system level or the hypervisor level, or the hardware level. In this project, we present a new type of rootkits called CloudSkulk, which is a nested virtual machine based rootkit. By impersonating the original hypervisor to communicate with the original guest OS and impersonating the original guest OS to communicate with the hypervisor, CloudSkulk is hard to detect, no matter whether defenders are at the higher-level (e.g., in the original guest OS) or at the lower-level (e.g., in the original hypervisor).**

I. Introduction

Over the years, it is commonly believed in the security community that the battle between attackers and defenders is determined by which side can gain control at the lower layer in the system [3]. Because of this perception, hypervisor-level defense is proposed to detect kernel-level rootkits [5], [7], [6], hardware-level defense is proposed to defend or protect hypervisors [1], [8].

In this project, we propose CloudSkulk - a nested virtual machine based rootkit that targets at a virtualized environment, in particular the cloud environment. The key feature of CloudSkulk is that the rootkit is inserted in between the original hypervisor and guest operating system (OS). Utilizing the nested virtual machine technique, the inserted rootkit in the middle (RITM) will impersonate the original hypervisor to communicate with the original guest OS, and meanwhile impersonate the original guest OS to communicate with the original hypervisor. Therefore, the presented rootkit is hard to detect, no matter whether defenders occupy the higher-level (e.g., in the original guest OS) or the lower-level (e.g., in the original hypervisor).

II. Design and Implementation

In this section, we describe the design and implementation of CloudSkulk. Our design and implementation is based on the Linux Kernel-based Virtual Machine (KVM) hypervisor. In a Linux system, the KVM hypervisor is implemented as two kernel modules (one architecture dependent and one architecture independent) of the host Linux system. Each virtual machine is then treated as a normal process, and is scheduled by the default Linux process scheduler. To create and launch virtual machines, users typically need to employ a user-level tool called Quick Emulator (QEMU). The rootkit we present depends on two techniques implemented in KVM and QEMU: nested virtualization and virtual machine live migration. Basically, there are five steps to install a CloudSkulk rootkit:

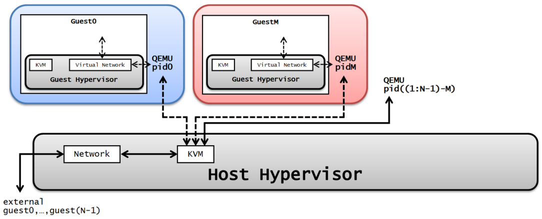

Fig. 1: Step 1 - Setup

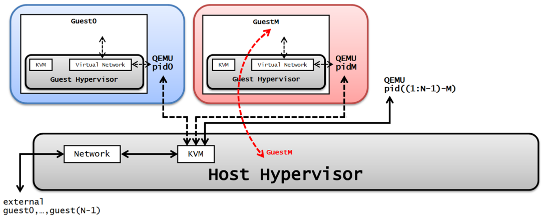

Fig. 2: Step 2 - Privilege Escalation

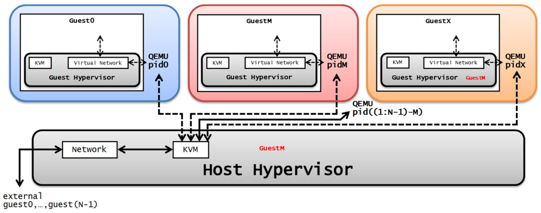

Fig. 3: Step 3 - Launch a New Virtual Machine

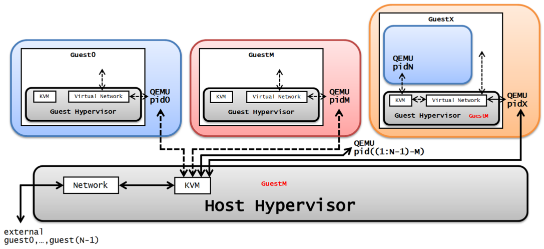

Fig. 4: Step 4 - Boot a Nested Virtual Machine

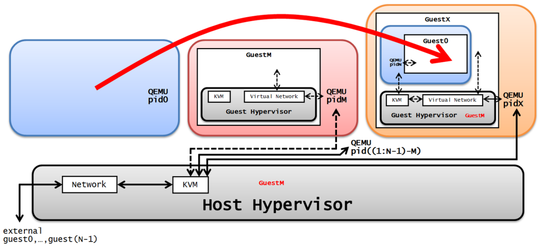

Fig. 5: Step 5 - Live Migration

- Step 1: Typically in a cloud environment, an attacker, just like normal cloud customers, can rent a virtual machine in the cloud environment. There could be many virtual machines co-existing on the same host machine as the attacker's virtual machine, and one of them, would be the target for attack. In Figure 1, we consider GuestM to be the virtual machine owned by the attacker, and Guest0 to be the target virtual machine.
- Step 2: We assume that by taking advantage of existing vulnerabilities in the hypervisor, the attacker is able to break out of its virtual machine and have some certain control on the host. This is feasible in reality as demonstrated in previous research [2], [4]. Note that the attacker does not necessarily need the system administrator privilege on the host, as a QEMU process can be launched by any normal user in a Linux system.
- Step 3: Once the attacker has some certain control on the host, the attacker can launch a new virtual machine - GuestX.
- Step 4: Utilizing the nested virtualization technique, the attacker can then launch a virtual machine inside GuestX.
- Step 5: Utilizing the virtual machine live migration technique, the attacker can migrate the target virtual machine (Guest0) to the nested virtual machine.

After the above five steps, the target virtual machine will be migrated into the new virtual machine as a nested virtual machine, and Guest0 will be running inside GuestX. At this moment, the attacker will kill the original virtual machine (as the source side of the migration).

III. Advantage of CloudSkulk

The major advantage of a CloudSkulk rootkit lies in its stealthiness. It is hard for both the virtual machine owner (i.e., the victim) and the system administrator to detect the existence of such a rootkit.

From the virtual machine owner's perspective, the owner does not observe any obvious behavior change. There are two reasons: on the one hand, when launching Guest0 and GuestX, port forwarding is used by the attacker, so that the victim will still be able to access its virtual machine using the same command as before; on the other hand, various techniques of detecting virtualization cannot be applied in this scenario, as the victim's machine is supposed to be running in a virtualized environment.

From the system administrator's perspective, GuestX will now be considered as Guest0. The attacker can ensure that GuestX and Guest0 are using the same OS, and run the same programs; meanwhile, with the complete control inside GuestX, the attacker has sufficient power to tamper with various virtual machine introspection (VMI) techniques.

IV. Experiments and Video Demonstration

To demonstrate the attack, we have taken a video, and the video is available on youtube: https://youtu.be/O2IMM52CCto.

In the video, we assume that the attacker has already gained some control on the host system. As shown in the video, the attacker does not need a system administrator's privilege, just a normal user's privilege would suffice to perform the attack, including launching virtual machines and initiating virtual machine live migration. The experiments are performed on a testbed running Fedora 22 operating system with Linux kernel 4.3.3. The guest OS is running a Fedora 25 workstation version, with Linux kernel 4.8.6. Our testbed platform uses Dell Precision T1700 with Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz processors. The Host has 16GB memory, and we assign each VM 4GB memory. It can be seen from the video that the time cost of the live migration is less than one minute.

V. Conclusion

In this project, we assume the perspective of an attacker and present CloudSkulk, a new type of rootkits that targets at cloud environments. Compared with existing rootkits, a CloudSkulk rootkit is stealthier as it tricks the two parties (the guest OS and the hypervisor) in a cloud to believe the rootkit is the other party.

References

[1] A. M. Azab, P. Ning, Z. Wang, X. Jiang, X. Zhang, and N. C. Skalsky, "Hypersentry: enabling stealthy in-context measurement of hypervisor integrity," in *Proceedings of the 17th ACM conference on Computer and Communications Security (CCS)*. ACM, 2010, pp. 38–49.

[2] N. Elhage, "Virtunoid: Breaking out of kvm," *Black Hat USA*, 2011.

[3] S. T. King and P. M. Chen, "Subvirt: Implementing malware with virtual machines," in *Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2006, pp. 14–pp.

[4] K. Kortchinsky, "Cloudburst: A vmware guest to host escape story," *Black Hat USA*, 2009.

[5] R. Riley, X. Jiang, and D. Xu, "Guest-transparent prevention of kernel rootkits with vmm-based memory shadowing," in *Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer, 2008, pp. 1–20.

[6] A. Seshadri, M. Luk, N. Qu, and A. Perrig, "Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses," in *Proceedings of twenty-first ACM SIGOPS Symposium on Operating Systems Principles (SOSP)*, vol. 41, no. 6. ACM, 2007, pp. 335–350.

[7] Z. Wang, X. Jiang, W. Cui, and P. Ning, "Countering kernel rootkits with lightweight hook protection," in *Proceedings of the 16th ACM conference on Computer and Communications Security (CCS)*. ACM, 2009, pp. 545–554.

[8] F. Zhang, K. Leach, K. Sun, and A. Stavrou, "Spectre: A dependable introspection framework via system management mode," in *Proceedings of the 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2013, pp. 1–12.