Executive Document for hyperjacking on vulnerable Type 1 and Type 2 hypervisors (revised December 2018)

Jay Chow, Kevin Hamilton, Yuqing Wang, Menghan Bai, Shivani Deshpande

Abstract - A hypervisor is a program that enables the hosting of different virtual machines on a single hardware component. It allocates resources to the VMs, each running a guest OS with applications. A fundamental component of virtualization but a single point of failure in security, the hypervisor could access all physical devices on a server, including memory and disk. There are 2 types of hypervisors, The Type is the bare metal also called native hypervisors and Type 2 being embedded also called as hosted hypervisors. For Type 1, the hypervisor runs on hardware to control and manage the VMs. This offers better performance for production purposes. For Type 2, the hypervisor runs as software on top of the OS as it needs a host OS to be installed on. As such, this is less secure. Hyperjacking involves installing a malicious hypervisor that could potentially take control of a server and steal data, maintaining persistence. This is highly dangerous as the VMs are not even aware that they have been compromised. To accomplish this task may pose difficulties as the attacker would need physical access to a server or install malicious code on the server. Additionally, a processor capable of performing hardware assisted virtualization may be needed.

1 Project Goals

The goal of our final project was to teach the class something new, interesting and applicable in the area of software vulnerability research. We have identified a topic relevant to software vulnerability analysis and provided materials to teach the class on this subject. Our research topic chosen was hyperjacking on vulnerable hypervisors. Hyperjacking interested us due to the wide range application that virtual machines are used today. As academics using virtual machines are powerful in testing software in specific operating systems. In production systems in the business world virtual machines are used to run critical services for operations. Based on the ubiquity of virtual machines it has become more and more common to see attacks via the white hat and black hat communities specifically targeting the strength of these systems. Our goal was to come out of this project with the ability to be able to speak intelligently about the past, present and future of hyperjacking threats.

1.1 Overview

Hyperjacking revolves entirely around the hypervisor so having a clear understanding of what a hypervisor is and does is important for the understanding of hyperjacking attacks. Hypervisors can be categorized into two different types, type 1 and type 2. For the purposes of this paper and experiment we focused solely on type-2 hypervisors. The reason for the focus on type-2 is that it is much more commonly used and the type used by the popular VMware and Virtualbox. Type-2 hypervisors serve as a middle layer between host OS and guest OS. The type-2 hypervisor creates and allocates virtualized hardware elements of the host machines. Type-2 hypervisors act as managers that isolate virtual machines from each other ensuring that problems in one virtual machines will not affect others being managed by the hypervisor. All the information sent from the hypervisor to the host operating system is formatted as if it is a normal host operating system process. This is necessary for completely different guest operating systems to run on a host operating system. This however poses a problem in traditional malicious behavior detection methods because hypervisors processes run lower than most host-based intrusion detection systems are looking.

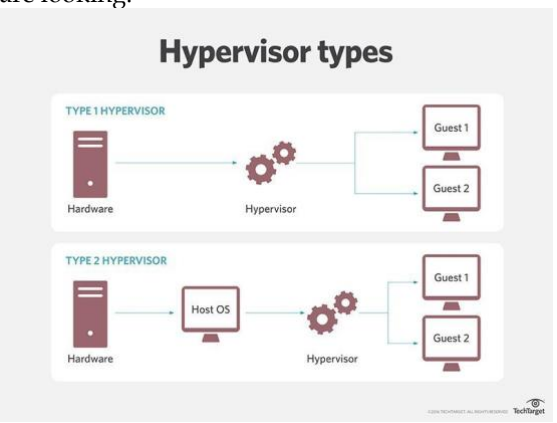

Fig. 1. Hypervisor Types (Type 1 and Type 2)cccccc

2 PROJECT EXECUTION

We studied about two specified hyperjacking vulnerabilities, CVE-2015-3456 and CVE-2012-5513, and tried to replicate them to gain a deeper understanding of the exploits. Venom, or CVE-2015-3456, is a famous QEMU vulnerability which was discovered by Jason Geffner. QEMU is an open source machine emulator which used for running virtual machines under the Xen and KVM/QEMU hypervisors. The vulnerability was revealed in May 2015 and might be the first vulnerability which had the potential to perform hyperjacking.

Affected Products and Versions	Patch Availability
VirtualBox 3.2, 4.0, 4.1, 4.2, 4.3 prior to 4.3.28	Oracle Linux and Virtualization
Oracle VM 2.2, 3.2, 3.3	Oracle Linux and Virtualization
Oracle Linux 5, 6, 7	Oracle Linux and Virtualization

Fig. 2. Affected Products and Versions by VENOM

VENOM is actually an out-of-bounds memory access flaw. It is found in the virtual Floppy Disk Controller (FDC) which handled FIFO buffer access. This flaw can be used to execute arbitrary code that can lead to hyperjacking or even crash the guest OS.

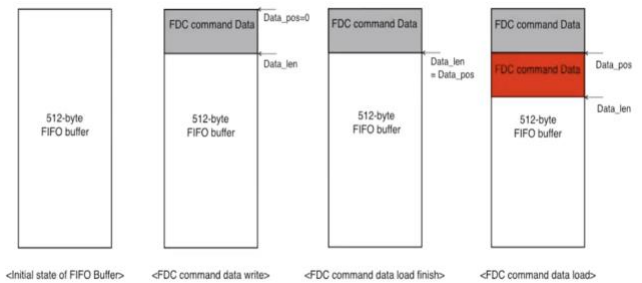

Fig. 3. Out-of-bounds memory access flaw in FIFO buffer

We found some codes which could exploit the VENOM on github and tested the payloads on Virtual Box Version 4.0.36. The operating system was Ubuntu 14.04 LTS which is vulnerable to VENOM attack according to the Ubuntu official website.

USN-2608-1: QEMU vulnerabilities

13 May 2015

qemu, qemu-kvm vulnerabilities

A security issue affects these releases of Ubuntu and its derivatives:

- Ubuntu 15.04
- Ubuntu 14.10
- Ubuntu 14.04 LTS
- Ubuntu 12.04 LTS

Fig. 4. Affected Ubuntu Versions

```
#include <sys/io.h>
#define FIFO 0x3f5
int main() {
        int i;
        iopl(3);
        outb(0x0a,0x3f5); /* READ ID */
        for (i=0;i<10000000;i++)
                outb(0x42,0x3f5); /* push */
}
```

Fig. 5. Payload to exploit the VENOM

For some reasons, we failed to crash the virtual machines in the test. We guessed that the vendor might have already patched the vulnerabilities in the platform we used since it is a severe flaw.

Another flaw we studied is CVE-2012-5513, related to XEN. This vulnerability is reported by Jann Horn in 2017. The attack can take control over the kernel of a Xen guest to break out of the hypervisor and futherly gain full control of the virtual machine's memory.

Xen guests share the virtual address with the hypervisor on x86-64. It allows the guest kernel to perform hypercalls, which are normal system calls from guest kernel t o hypervisor. The hypercalls often take guest pointers as arguments to pass the pointers. To ensure guest-virtual pointers don't point to hypervisor-owned memory, it uses a user space accessor to check the pointer .

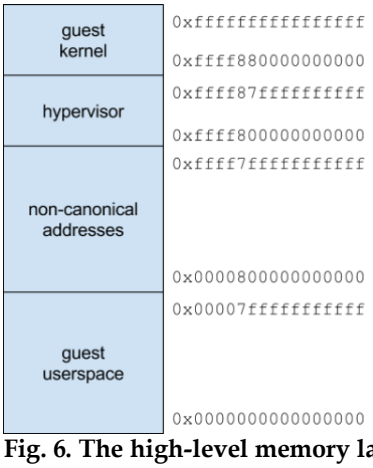

Fig. 6. The high-level memory layout of Xen

The hypercall can invoke the function memory_exchange (arg) in xen.c which allows a guest trade in a list of physical pages in exchange for some new pages with different restrictions on physical contiguity. However, this function did not check the userspace array pointers before accessing them, so it was possible to cause Xen to read from and write to hypervisor memory.

3 ACCOMPLISHMENTS

Hyperjacking is still in its infancy and still only a theoretical attack scenario. There has not been any report of an actual demonstration as on early 20015, of a

successful Hyperjacking in the wild. Hyperjacking is quite considering that it is difficult to directly access hypervisors but it is still considered as a real-world threat and it has the potential of serious damage in case it is successfully exploited.

One of our accomplishments was being exposed to Hyperjacking-style threats such as Blue Pill, SubVirt and Vitriol . Blue Pill is a malware based on virtualization. It can run as a hypervisor between the host OS system and the hardware. The malware is claimed to be undetectable because it does not use system hooks. So, the rootkit detection method is no longer working. SubVirt is a virtual machine-based rootkit. It attacks through attacking the operating system and makes the malware functions as the root hypervisor.

Vitriol is capable of transforming a running operating system into a 'hardware virtual machine' and load itself as a 'rootkit hypervisor'. Virus scanners and rootkit tracers become helpless in protecting systems against such rootkits.

There are various ways to carry out a hyper-jack such as VM Escape, VM Hopping, VM Theft and VM Sprawl. However, even though these exploits are possible, they are not, as far as we know, a practical reality, since any effective intrusion prevention system make this sort of attack difficult.

We tried to exploit the VENOM vulnerabilities in our tests. According to the official websites, VirtualBox 4.0 should be a vulnerable platform. We also found some information about vulnerable version Ubuntu on their websites. The specific environment we used is: Oracle VirtualBox version 4.0.36 & Ubuntu 14.04 LTS.

Fig. 7. Environment for testing VENOM

However, we did not succeed in crashing our virtual machine in this test. Although we used the vulnerable platform and operating system, the qemu installed on the Ubuntu we used might have been already patched. Since VENOM was rated as having an important impact, it is difficult to find a vulnerable version without patching on the Internet.

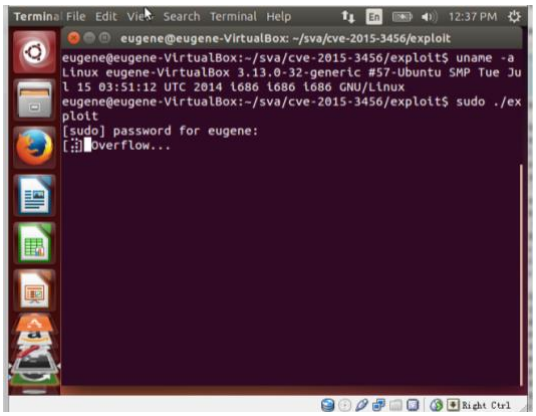

Fig.8. Attempt to exploit VENOM

4 LESSONS LEARNED

We learned that the detailed differences between Type 1 and Type 2 hypervisors. For example, Type 1 hypervisors work on the hardware of the host and can monitor guest OSes in the VMs above it. In this case, the hypervisor is completely independent from them and the main task of this hypervisor is to manage the sharing of resources between the different VMs. There is an advantage here as complications in a VM should not spill over to other VMs running on the common hypervisor. In contrast, Type 2 hypervisor is installed on the host OS and is there completely reliant on the host OS. In terms of security, this may be less secure because vulnerabilities in the base OS will affect the hypervisor running on top of it. The difference between the 2 types is the presence of an underlying host OS.

Additionally, we learned the vulnerability of CVE-2015-3456 in greater detail. For the former, as a group, we learned a great deal about the VENOM vulnerability in virtual floppy drive code. The consequences are severe as this could give an attacker elevated access to the host machine's local network, exposing access to corporate intellectual property, personally identifiable information which could impact many organizations and users that utilize VMs in real-world situations. The venom vulnerability can be depicted pictorially below:

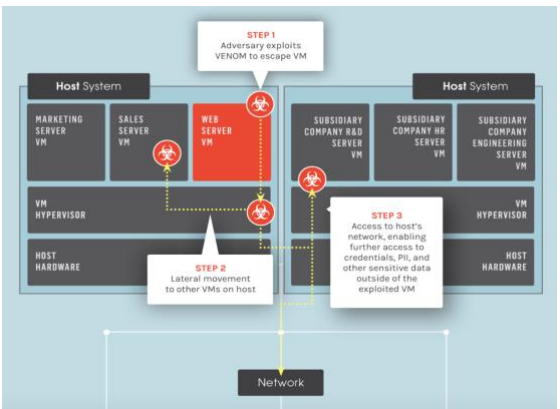

Fig. 9. Adapted from ttps://venom.crowdstrike.com/

This vulnerability allows attackers with root privileges to escape Guest machine (VMs) and allows the access to the host system. To successfully exploit this vulnerability, an attacker needs permissions to access the floppy disks controller(FDC). On the guest VM running Linux, the attacker needs root or elevated privileges. On the guest VM running Windows, anyone could have permissions to access the FDC. Some cloud providers running previous versions of Red Hat Enterprise Linux which relied heavily on virtualization were vulnerable to venom. To fix this, the users of Red Hat had to update their systems using "yum update" or "yum update qemu-kvm".

Since the XENMEM_exchange handler in Xen 4.2 actually does not check the memory address, and this means the guest OS administrators may gain privileges through vectors that overwrite the memory in the hypervisor or even attempt a system crash. More specifically, XENMEM_exchange handler can access guest memory without checking the range of guest provided addresses which allows the accesses the hypervisor reserved range. This is caused by improper input validation. To exploit, a guest user c ould crash Xen or in rarer cases, cause privilege escalation.

We also learned about some of the security practises that must be followed in a virtualized environment. Hypervisors should be patched as and when the security upgrades are released by vendors to ensure that there are no open vulnerabilities that can be exploited by attackers. Appropriate access control mechanisms must be implemented to ensure Guest OS does not have access to Host OS. This will ensure that even in case of compromise of guest OS, attacker will not be able to take control of the base server.

Although virtualization has broadened the horizon for faster computing and has pushed the performance boundaries from having just a single OS to enabling a thousands of OS running at the same time but it has also expanded the threat landscape and there still are many risks that must be addressed with much forethought. Hypervisors have brought increased convenience and scalability to users, but at the expense of introducing security vulnerabilities due to the increased complexity of the code.

5 References

[1] Vincent Bernat, "Experiments related to CVE-2015-3456". May 27, 2015. Available: https://github.com/vincentbernat/cve-2015-3456?fbclid=IwAR3ySVhZFqQaa1x0TiP0DWVjjdldaZz7eTYurw_jp0i6aEJqj88KjHsugJ4.

[2] WEBTECHMAG, "Free Virtualization Software & Hypervisors". Available: http://webtechmag.com/free-virtualization-software-hypervisors/.

[3] Dimitri McKay, "A Deep Dive Into Hyperjacking". February 03, 2011. Available: https://www.securityweek.com/deep-dive-hyperjacking

[4] Wikipedia, "Hyperjacking". November 11, 2017. Available: https://en.wikipedia.org/wiki/Hyperjacking

[5] International Journal of Engineering Research and Modern Education (IJERME) ISSN (Online): 2455. – 4200 (www.rdmodernresearch.com) Volume I, Issue I, 2016

[6] David Marshall, "Hyperjacking - the latest threat to servers". May 24, 2007. Available: http://vmblog.com/archive/2007/05/24/hyperjacking-the-latest-threat-to-servers.aspx#.XArnoBNKib8

[7] Security Impacts of Virtualization on a Network Testbed", Yu-Lun Huang, Borting Chen, Ming-Wei Shih, Chien-Yu Lai. 2012. Page 5. IEEE Sixth International Conference on Software Security and Reliability.

[8] Harold F. Tipton, Micki Krause Nozaki. Information Security Management Handbook, Volume 4. CRC Press, June 3, 2011. Virtualization Malware. Page 529.

[9] Samuel T. King Peter M. Chen. "SubVirt: Implementing malware with virtual machines". University of Michigan. May 2006. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/subvirt.pdf

[10] SPAMfighter News, "The Vitriol VM Rootkit is Unguarded". October 25, 2006. Available: https://www.spamfighter.com/News-6634-The-Vitriol-VM-Rootkit-is-Unguarded.htm

[11] Pearce, M., Zeadally, S., and Hunt, R. 2013. Virtualization: Issues, security threats, and solutions. ACM Comput. Surv. 45, 2, Article 17 (February 2013), 39 pages.

[12] Andrew Clarke, "Towards Security as a Service?".26 March 2013. https://www.red-gate.com/simple-talk/blogs/towards-security-as-a-service/

[13] Michael Gabriel Sumastre, "Virtualization 101: What is a Hypervisor?". February 27, 2013. Available: https://www.pluralsight.com/blog/it-ops/what-is-hypervisor

[14] "Towards Security as a Service?", Andrew Clarke. 26 March 2013. https://www.red-gate.com/simple-talk/blogs/towards-security-as-a-service/

[15] https://profsandhu.com/cs6393_s14/csur_virt_2013.pdf

[16] https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8419241

[17] https://venom.crowdstrike.com/

[18] https://www.eventtracker.com/blog/2015/june/venom-vulnerability-exposes-most-data-centers-to-cyber-attacks/

[19] https://lists.opensuse.org/opensuse-security-announce/2012-12/msg00000.html