ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 9, September 2023

Overview of Virtualization, Attacks, Different Case Studies and Mitigations

[1] Naga Sesha Venkata Pavan Kumar Kavuluru, [2] Ramakrishna Vankamamidi, [3] Sai Vineeth Goka

[1] [2] [3] Network Security-ERS, HCL Technology LTD.,
Corresponding Author Email: [1] kavuluru.p@hcl.com, [2] rvankama@hcl.com, [3] goka.saivineeth@hcl.com

Abstract*— Cloud computing is the most trending topic in today's IT world, and one of its key technologies is virtualization. Virtualization enables us to create useful environments from abstract resources by separating functions from the underlying hardware. However, as cloud computing usage has increased, so as the threats to its various security layers, including the virtualization layer. Attackers have increasingly targeted this layer with malicious activity, with the potential for compromising VM infrastructures leading to access to other VMs on the same system and even the host. In this paper, we have highlighted the different types of threats that can compromise the virtualization layer. The emerging VM escape attack is particularly concerning among the several types of attacks that can occur in the virtualization layer [1]. If attackers gain control of the Virtual Machine Monitor (VMM), they will have full control of all VMs and accessed data, as well as the underlying physical system and hosted applications. In this paper, we have analyzed the use of the Bell-LaPadula model as a base method for implementing the PVEM model as a mitigation strategy. Furthermore, we also cited the bounds-check bypass attack, one of the techniques for VM escape. With the current analysis, it seems that there are still potential areas to address this issue and we aim to do further research in this area.

***Keywords*—cloud security, virtualization layer, virtual machine escape.**

I. BACKGROUND

Virtual Machine Escape Attack

A virtual machine escape attack is a type of security breach in which an attacker exploits vulnerabilities in the hypervisor to gain unauthorized access to other virtual machines on the same server. The goal of the attacker is to bypass the isolation provided by virtualization and gain access to the host system or other virtual machines, which can result in the theft of sensitive data and other security breaches. This type of attack can pose a significant threat to the security of virtualized environments, as it allows an attacker to gain access to resources and authorities that they should not have[2].

Virtual machine escape attacks can be carried out using various techniques, including bypassing bound checks, buffer overflow, code injection attacks, and privilege escalation attacks[3].

II. HOW IT IS IMPACTING THE CLOUD

If a VM escape attack is successful at the IaaS layer, it can cause great damage as it can potentially compromise the entire technology stack, including the layers built on top of it (PaaS and SaaS).

When an attacker gains access to the IaaS layer through a VM escape attack, they can potentially compromise the entire technology stack. This can include accessing sensitive data, manipulating, or stealing data, disrupting critical services, and even launching additional attacks on the PaaS and SaaS layers. Furthermore, the attacker can potentially move laterally within the cloud infrastructure and infect other virtual machines.

III. WHY VIRTUAL MACHINE ATTACK IS IMPORTANT TO ADDRESS?

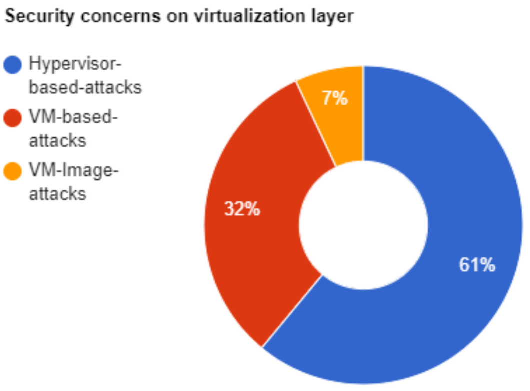

Figure 1. Security concerns on virtualization.

As we can see from the above Fig. 1. that hypervisor-based attacks are more compared to other attacks in the virtualization layer. Virtual machine escape attack comes under the hypervisor-based attack [3].

Cloud computing security and privacy issues have been extensively studied, particularly in the virtualization layer. Virtualization security issues cover topics such as virtual image management, monitoring, network virtualization, mobility, and malware. The virtual machine monitor is a crucial aspect of virtualization security. The Virtual Machine Monitor (VMM) is a critical software component that manages and isolates each running virtual machine (VM). However, VMM can have many entry points and interconnection complexities, which can increase the number of potential attack vectors.

IV. VM ESCAPE ATTACK ANALYSIS

Intel's Virtualization Technology is a hardware-assisted virtualization solution that allows x86 processors to run virtual machines with the help of a hypervisor.

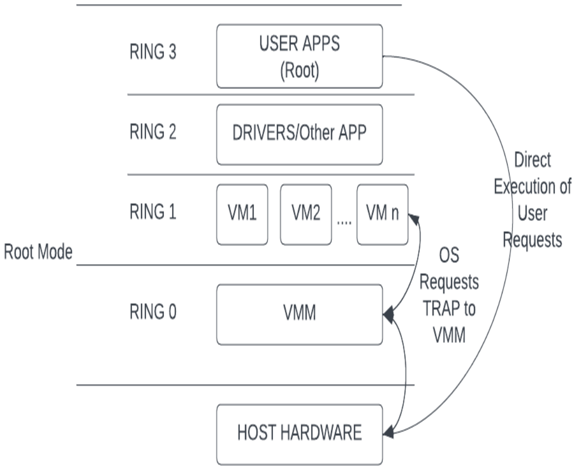

Figure 2. Hardware-assisted full virtualization.

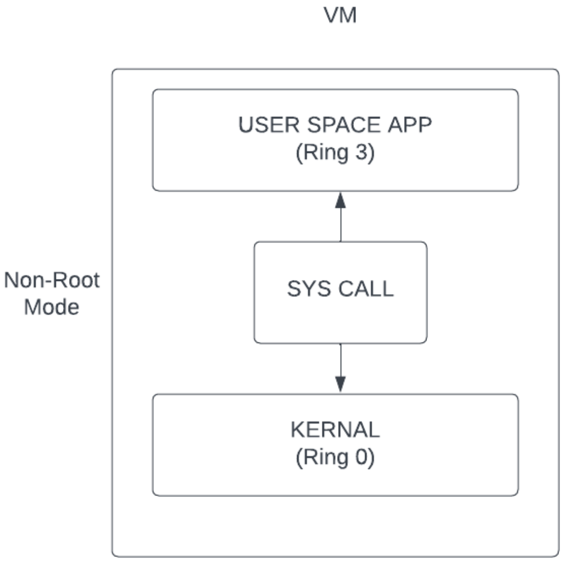

Figure 3. Virtual machine non-root mode.

In a virtualized environment, the operating system kernel must have the highest level of control, known as Ring 0, to manage the computer's resources effectively. However, granting the same level of control to both the hypervisor (which manages virtual machines) and guest operating systems can lead to instability and chaos in the system. To address this, a technique called Ring Deprivileging is used. It involves running the guest operating systems at a lower level, either Ring 1 or Ring 3, while the hypervisor maintains its control at Ring 0 in the root mode [2].

A Virtual Machine Escape Attack is a dangerous security breach where an attacker uses malicious applications to gain the highest level of access in a virtual machine. In a hardware-assisted full virtualization setup (as shown in Fig. 3), an attacker aims to move from Ring 1 of non-root mode privilege to Ring 0 of non-root mode, which gives them complete control over the virtual machine. Once the attacker achieves this, they can perform any operation allowed by the hypervisor, including accessing sensitive data, manipulating information, stealing data, and even disrupting critical services [4].

As illustrated in Fig. 2, the attacker interacts with the hypervisor using I/O control analog commands during a virtual machine escape attack. By simulating pseudo-I/O operations, the attacker gains Ring 1 of root mode privilege. From there, they can exploit vulnerabilities in the hypervisor or inject malicious code into it. Once the attacker gains Ring 0 of root mode privilege, the hypervisor and the host operating system become vulnerable. This means that the attacker can compromise the data and running states of other virtual machines on the host, potentially causing extensive damage to the entire technology stack [4].

To protect against such attacks, it is essential to develop robust security measures and carefully manage the interactions between virtual machines and the hypervisor. Detecting and addressing potential vulnerabilities in the hypervisor becomes crucial in preventing malicious actors from gaining unauthorized access and disrupting the integrity of the cloud environment. Through continuous research and collaboration, the cloud community can work together to strengthen security measures and ensure the safe and reliable functioning of virtualized systems.

V. VM ESCAPE ATTACK

The process of a VM escape attack involves two main steps, placement and extracting information. The attacker first needs to place their malicious virtual machine on the same physical machine as the target host or hypervisor. This process is challenging because it requires the attacker to bypass various co-residency detection methods cloud service providers use. Once the attacker successfully places their malicious virtual machine on the same physical machine, they can launch attacks to extract information from other virtual machines on the same host [2][6].

VI. HOW DO ATTACKERS APPROACH?

An attacker must overcome three main challenges to conduct a successful VM escape attack in a cloud environment. First, the attacker cannot determine whether their malicious virtual machine is co-resident with the target host or hypervisor they want to attack, making placement difficult. Second, the attacker must identify any security vulnerabilities on the host or hypervisor that they can exploit to gain key permissions. Finally, the attacker needs to know how to extract critical information from other virtual machines through the escape attack once they have successfully deployed their malicious virtual machine [2].

VII. CASE STUDY

1) CVE-2008-0923

A vulnerability (CVE-2008-0923) in VMware discovered by Core Security Technologies made VM escape possible on VMware Workstation. The vulnerability was found in VMware's shared folders mechanism, By exploiting this vulnerability, the Guest system gains unauthorized access to and control over any folder, including the system folder and other security-sensitive directories, of the Host system, thereby compromising its security.

Attackers used a specially crafted sequence of p-trace system calls to modify the memory space of a target process. By doing so, the attacker gained root access to the system and executed arbitrary code.

a) BELL-La PADULA (BLP) MODEL

In the BLP model, a simple security property is a fundamental rule that governs the reading of information. It states that a subject can only read an object if the security level of the subject is greater than or equal to the object's security level. This means that a subject with a lower security level cannot read or access an object with a higher security level. For example, a user with a "confidential" security clearance cannot read a file marked as "top secret" unless they are also granted access to that security level [2].

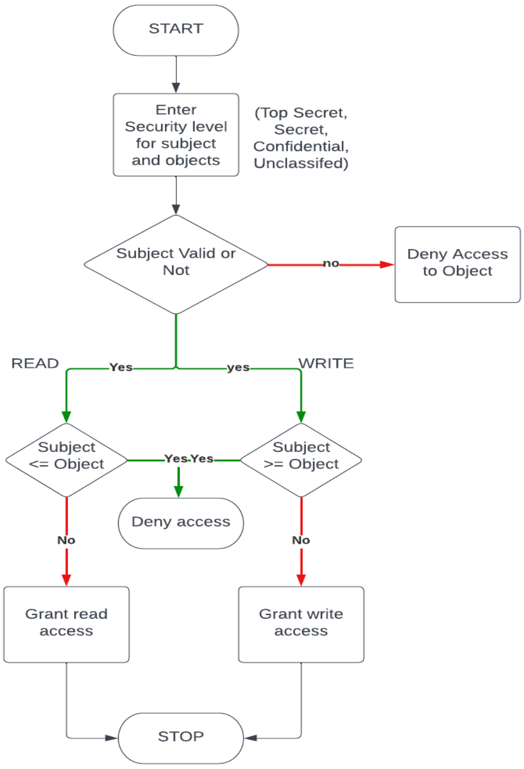

Figure 4. Flow chart of BLP model.

On the other hand, a *-property is a rule that governs the writing of information. It states that a subject can only write to an object if the security level of the subject is less than or equal to the object's security level. This means that a subject with a higher security level cannot modify or write to an object with a lower security level. For example, a user with a "top secret" security clearance cannot write to a file marked as "confidential" unless they are also granted access to that security level.

Together, the simple security property and the *-property form the basis of the BLP model's security policy. The model enforces a mandatory access control (MAC) policy, where access to objects is controlled by the security levels of the subjects and objects. This ensures that sensitive information is protected from unauthorized access, modification, or disclosure, and helps to prevent data breaches or leaks caused by insider threats or external attacks.

b) PVME MODEL

The PVME model consists of four parts: PVME_Hook, Running Model, Visit Matrix, and Learning Matrix. As shown in Fig. 5. It operates in two modes: Learning mode and Enforce mode. In Learning mode, all VM operations are recorded in the Learning Matrix for analysis and deduplication. The results are then added to the Visit Matrix. The PVME_Hook function receives messages from QEMU and checks their compliance against the Visit Matrix. If compliant, the Hypervisor's Sys_Call is returned. If not, the PVME model's running mode is checked. If the running mode is Enforce, an Error is returned and the request is rejected. In Learning mode, the request is logged in the Learning Matrix and the Sys_Call is returned. Overall, the PVME model ensures compliance and allows for collecting noncompliant operations to improve its versatility [2].

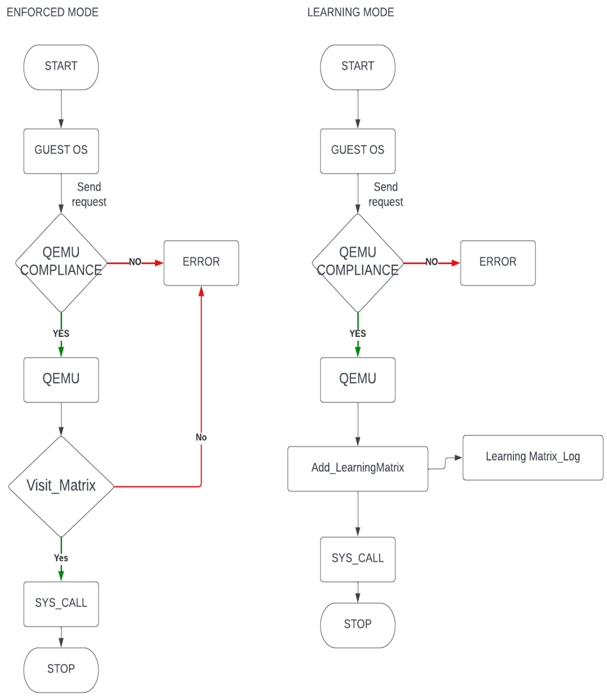

Figure 5. Flow chart of PVME model

2) CVE-2017-5753

In 2017, in the Xen hypervisor a team of researchers discovered a VM escape vulnerability that allowed an attacker to escape from a guest VM and gain access to the host system. This vulnerability, highlighted through CVE-2017-5753 or "Xen Project Security Advisory 254," is the result of a flaw in the way the Xen hypervisor handles certain instructions when a guest VM is running in 64-bit mode [5].

"Also Bounds-check bypass is another variant of attack that can exploit the behaviour of modern processors. This was highlighted via CVE-2017-5753. It involves manipulating the branch predictor to deceive the processor into speculatively executing code beyond established security boundaries and checks. This speculative execution can enable an attacker to execute code with unpredictable array indexes, potentially compromising the security of the system.

A branch predictor is a hardware component that helps the processor predict which instructions to execute next in a program. The attacker aims to "poison" the branch predictor by feeding it with misleading or incorrect information, causing it to mis-predict the path that the program should follow.

As a result of this attack, the victim code, which is the code being executed by the processor, may be speculatively executed past its intended boundaries and security checks. Speculative execution is a processing technique that predicts and executes instructions ahead of time to enhance program execution speed. However, in this case, the predicted instructions may be incorrect due to the poisoned branch predictor, leading to unexpected behaviour.

One potential consequence of this attack is that speculative code in the normal hyper call/emulation path may execute with wild array indexes. In other words, the attacker may be able to execute code that accesses memory locations outside the boundaries of an array, potentially causing memory corruption and other security issues. This can be particularly dangerous in the context of virtualization and emulation, where the attacker may be able to escape the virtual environment and access the host system.

On January 3, 2018, it was publicly disclosed that this security vulnerability in modern processors could allow the software to exploit CPU data cache timing to leak information and potentially lead to unauthorized access to virtual memory. This vulnerability, known as Spectre and Meltdown, has three variants that can affect processors from Intel, AMD, and ARM.

Variant 1, also known as Spectre (CVE-2017-5753 and CVE-2018-3693), involves bypassing bounds checks and allows attackers to read sensitive data from other parts of the system.

Variant 2, also called Spectre (CVE-2017-5715), uses branch target injection to manipulate the processor's branch predictor and trick it into executing speculative code that can access protected information.

Variant 3, known as Meltdown (CVE-2017-5754), exploits the way modern processors optimize data caching and allows attackers to read data from the kernel and other processes they should not have access to.

To mitigate these vulnerabilities, all components from the operating system to the CPU microcode must be patched or updated. Operating systems are implementing various mitigations to reduce the attack surface, and some will be more effective when the CPU microcode is also updated. However, applying these mitigations may cause a performance impact. It is crucial for users to ensure their systems are up-to-date with the latest security patches to protect against these vulnerabilities.

Overall, these type of attacks highlights the importance of securing the hardware components of a computer system to prevent malicious attackers from exploiting vulnerabilities.

VIII. CONCLUSION

In conclusion, virtual machine escape attacks pose a significant threat to the security of cloud computing environments, with the potential to compromise the entire technology stack and expose sensitive data. As cloud adoption continues to grow, it becomes imperative to address the vulnerabilities within the virtualization layer. The Bell-LaPadula model and the PVME model have been proposed as potential mitigation strategies, but further research is needed to assess their effectiveness in real-world scenarios. Additionally, the analysis of specific vulnerabilities, such as the bounds-check bypass attack, underscores the importance of continuously evaluating and securing hardware components to prevent exploitation. Moving forward, collaborative efforts between industry, academia, and cloud service providers will play a vital role in developing robust defense mechanisms and best practices to safeguard against virtual machine escape attacks. By focusing on practical implementation, case studies, emerging technology considerations, and standardized security protocols, we will try to build a more secure and resilient cloud ecosystem, ensuring the protection of critical data and services from ever-evolving threats.

IX. FUTURE RESEARCH

With the current analysis, it seems that there are still potential areas to address the Bounds-check bypass issue and VM Escape attack, we aim to do further research in this area [7].

We are working on finding better ways to protect the cloud from virtual machine escape attacks. We will try to explore combining different security methods to create a strong defense against these attacks. Using advanced techniques like machine learning and behavior analysis can help detect and stop attacks more effectively. Regularly checking hardware security features and fixing any weaknesses will be important to keep virtualized systems safe. Focusing on these areas can make the cloud more secure and ensure our data and services stay protected from virtual machine escape attacks.

REFERENCE

[1] Security Issues in Distributed Computing System Models

[2] An Access Control Model for Preventing Virtual Machine Escape Attack by Jiang Wu, Zhou Lei, Shengbo Chen, and Wenfeng Shen.

[3] Virtualization vulnerabilities, security issues, and solutions: a critical study and comparison.

[4] Intel VMX technology G. Lettieri 28 Oct. 2015.

[5] https://xenbits.xen.org/xsa/advisory-254.html.

[6] Security Architecture of Cloud Computing by V. Krishna Reddy et al. / International Journal of Engineering Science and Technology (IJEST).

[7] Spectre (security vulnerability) - Wikipedia.

[8] Grobauer, B.; Walloschek, T.; Stocker, E. Understanding cloud computing vulnerabilities. IEEE

[9] Modi, C.; Patel, D.; Borisaniya, B.; Patel, A.; Rajarajan, M. A survey on security issues and solutions at different layers of Cloud computing.

[10] Islam, T.; Manivannan, D.; Zeadally, S. A classification and characterization of security threats in cloud computing.

[11] Han, Y.; Chan, J.; Alpcan, T.; Leckie, C. Using Virtual Machine Allocation Policies to Defend against Co-Resident Attacks in Cloud Computing.

[12] Liu, Q.; Wang, G.; Weng, C.; Luo, Y.; Li, M. A Mandatory Access Control Framework in Virtual Machine System with Respect to Multilevel Security II: Implementation.

[13] Zhu, H.; Xue, Y.; Zhang, Y.; Chen, X.; Li, H. V-MLR: A Multilevel Security Model for Virtualization. In Proceedings of the International Conference on Intelligent Networking and Collaborative Systems.

[14] Xue, H.; Zhang, Y.; Guo, Z.; Dai, Y. A Multilevel Security Model for Private Cloud.

[15] Yan, Z.; Li, X.; Wang, M.; Vasilakors, A.V. Flexible Data Access Control based on Trust and Reputation in Cloud Computing.

[16] VMware Response to Speculative Execution security issues, CVE-2017-5753, CVE-2017-5715, CVE-2017-5754, and CVE-2018-3693 (aka Spectre and Meltdown) (52245)

[17] Singh, A.; Chatterjee, K. Cloud security issues and challenges: A survey. J. Netw. Comput. Appl. 2017, 79, 88–115.