שאלות תיאורטיות

1. (א) יצרנית המכונות־הווירטואליות Broadcom הודיעה בתחילת מארס על פרצת אבטחה בתכנת ה־Type 1 Hypervisor הנפוצה שלה, VMWare ESXi:

> "On March 5, 2024, Broadcom issued a security advisory, VMSA-2024-0006, which addresses security vulnerabilities discovered in VMware ESXi, VMware Workstation Pro & Player, and VMware Fusion. An attacker with privileged access (root or administrator) to the guest OS inside a virtual machine may use these vulnerabilities to access the hypervisor." (https://core.vmware.com/resource/vmsa-2024-0006-questions-answers#introduction)

תאר בקצרה את פרצת האבטחה הזו במושגים שלמדנו. פרצת האבטחה הזו גרמה לבהלה בקרב חברות ענן רבות. הסביר מדוע ותן דוגמה לתרחיש קונקרטי שמפחיד אותן בשל הפרצה הזו.

(ב) אם הבעיה הייתה בתכנת Type 2 Hypervisor, האם זה היה מטריד חברות ענן באותה מידה? מדוע?

(ג) בכתבות על הפרצה הזו, נכתב שמדובר בפרצה מסוג "out-of-bounds write":

> "A malicious actor with privileges within the VMX process may trigger an out-of-bounds write leading to an escape of the sandbox" (https://www.itnews.com.au/news/vmware-patches-against-sandbox-escape-605834)

המונח הזה פירושו ניסיון לכתוב למקום בזיכרון שלא הוקצה. לדוגמה, `example-list=[1, 2, 3]` ואז `example-list[7]=5` (הדוגמה הזו לא תעבוד בפייתון, כי היא מונעת שימושים לא־חוקיים כאלו, אבל בשפות־תכנות אחרות זה אפשרי). האם סוג הפרצות האלו מתרחש בזיכרון בערימה או במחסנית? מדוע כן ומדוע לא?

האם כתיבה מסוג out-of-bounds write על איברים בערימה יכולה לדרוס מידע שנמצא במחסנית?

(a) This security breach means that an attacker who has compromised a virtual machine can exploit these vulnerabilities to gain unauthorized access to the underlying hypervisor, potentially compromising the security and integrity of the entire virtualized environment.
Type 1 hypervisor attack is risky for cloud companies for a few reasons: first, Type 1 hypervisor sits directly on the physical hardware of a server and controls the allocation of hardware resources to virtual machines. So, if an attacker gains control over the hypervisor, they essentially gain control over the entire infrastructure. This will enable the attacker to access all VM's and potentially extract sensitive data or disrupt operations.
Another reason is the fact that in cloud companies like Amazon for example, multiple customers share the same physical infrastructure. A vulnerability in the hypervisor could allow an attacker to breach the isolation barriers between VMs, giving them unauthorized access to data from multiple customers. This violation of data segregation not only compromises the confidentiality and integrity of customer data but also defects the trust between the cloud provider and its customers.

Lastly, Many industries are subject to strict regulations governing data security and privacy. A hypervisor vulnerability that leads to a data breach could result in severe penalties, legal liabilities, and reputational damage for the cloud provider.
An extremely alarming scenario arising from such an attack might involve a sophisticated threat actor exploiting the hypervisor vulnerability to infiltrate a cloud provider's infrastructure. Once inside, they could stealthily move across VMs, exfiltrating data from multiple customers, including personally identifiable information, financial records, or intellectual property. In a worst-case scenario, the attacker could manipulate critical infrastructure services, causing widespread service disruptions or even complete system outages. The resulting fallout could lead to substantial financial losses, legal liabilities, and irreparable harm to the affected businesses and their customers.

(b) If the problem were in a Type 2 Hypervisor program, it might not trouble cloud companies to the same extent. Type 2 Hypervisors run on top of a host operating system, which adds an additional layer of isolation compared to Type 1 Hypervisors. While still concerning, a vulnerability in a Type 2 Hypervisor might not directly expose the underlying infrastructure to the same extent as a Type 1 Hypervisor vulnerability. Cloud companies could potentially mitigate the impact more easily by patching or isolating affected virtual machines.
Also, since there is no access to the physical hardware, the attacker won't be able to take control over the whole system.

(c) Out-of-bound write vulnerabilities can indeed occur on both the heap and the stack. However, implementing OOBW on the stack is generally considered easier due to certain characteristics of stack memory management. The stack typically has a fixed size, making it easier for an attacker to predict the exact boundaries of allocated memory regions.
As a result, it's relatively simpler for attackers to craft malicious input that overflows stack-allocated buffers and overwrites adjacent memory, such as return addresses, leading to a stack-based buffer overflow.

In contrast, exploiting OOBW vulnerabilities on the heap can be more complex due to the dynamic nature of heap memory management.
In addition to the allocated data, heap memory typically contains metadata that store information about allocated memory, such as the sizes and pointers to neighboring blocks. While this metadata can potentially be manipulated by attackers, it adds an additional layer of complexity to crafting exploits compared to stack-based vulnerabilities. Attackers may need to carefully manipulate heap metadata to trigger specific heap layout conditions and achieve out-of-bound writes beyond the allocated heap space.
Overall, while both heap and stack memory are susceptible to OOBW vulnerabilities, exploiting such vulnerabilities on the stack is often considered more straightforward due to the deterministic nature of stack memory allocation. However, sophisticated attackers may still leverage weaknesses in heap memory management to craft exploits for OOBW vulnerabilities.