Activity 8: Part 1

Ryan Coon

CST-407

Professor Ty Howard

February 11, 2024

Operating Systems Security

Enforced resource sharing refers to the mechanism by which the system ensures fair and efficient allocation of resources among competing processes or users. In essence it ensures that all processes or users have equal access to resources available such as CPU time, memory, disk space, and even network bandwidth.

Guaranteed fair service is the assurance provided by the system that all users or processes will receive a fair share of the system resources within a specific time frame. This is especially important in multi-user environments where multiple processes compete for resources.

Memory protection is important to multi user environments as well. It refers to the mechanism by which the system will safeguard memory space assigned to each process or application. The operating system will then ensure that each process or application operates within its own memory space, preventing one process from accessing or modifying memory allocated to another process. This is crucial for preventing data corruption and unauthorized access.

A “fence” involves the use of special hardware to establish a boundary or fence between different memory regions. This is designed to enhance security and prevent unauthorized access to specific memory areas. This method is performed by using hardware mechanisms such as paging or segmentation.

Boundary registers involve the use of special hardware registers to define the boundaries of memory regions for processes or tasks. In essence it adjusts memory allocation based on system needs. On a multiple user system, allocated memory will change the memory boundary based on the need and then protect the memory from being modified. This solves wasted space problems, but a user can accidentally store data on top of instructions and possibly destroy programs or overwrite memory.

Tagged memory is a development  that involves associating metadata or tags with memory locations to provide additional information about the contents of those memory areas. The rationale behind the development of tagged memory is driven by the need to enhance security, facilitate memory protection, and support more efficient memory management within an operating system.

Segmentation involves dividing the primary memory into variable-sized partitions or segments, each of which is dedicated to a specific process or task. The rationale behind the development of segmentation is driven by the need to support flexible memory allocation, facilitate multitasking, and enable efficient memory utilization within an operating system.

Paging involves dividing the physical memory into fixed-size blocks called "pages" and organizing the logical memory of a process into equally sized "frames." The rationale behind the development of paging is driven by the need to facilitate efficient memory allocation, support virtual memory systems, and simplify address translation within an operating system.

Layered privilege processes, also known as layered trust, is an operating system security feature that involves dividing a system into distinct layers or levels of privilege, with each layer having specific access rights and capabilities. This approach is designed to enhance security by minimizing the potential impact of security breaches and limiting the scope of unauthorized access.

Virtualization is an operating system feature that enhances security in several ways, providing isolation, flexibility, and control over system resources. In other words this is a fully functioning simulation of an operating system that allows the user to test and run programs without the worry of causing damage to the actual operating system.

A sandbox is very similar to virtualization, where it creates a protected environment where a user can run untrusted or potentially malicious programs without spreading or causing harm to the actual operating system. In other words it isolates the execution of applications or processes from the rest of the system, preventing them from accessing sensitive resources or modifying critical system components. This isolation helps contain the impact of security breaches, as any compromise within the sandbox is limited to that specific environment.

A rootkit is a type of malicious software designed to gain unauthorized access to a computer or system while remaining hidden from detection. Rootkits are often used by attackers to maintain long-term control over compromised systems, allowing them to carry out unauthorized activities without the knowledge of the system's legitimate users or administrators. One way a rootkit can remain hidden is using a kernel level rootkit. By operating at the kernel level, the rootkit then has the ability to intercept system calls and manipulate data at a deep level. Therefore by modifying the behavior of the operating system’s kernel, these rootkits can effectively conceal their presence from traditional security software and monitoring tools.

An anti-rootkit tool is used to detect, identify, remove, and protect against rootkit-based threats by scanning the system for anomalies, uncovering rootkit artifacts, removing malicious components, preventing future infections, verifying system integrity, and providing user guidance and reporting. By leveraging advanced detection and mitigation techniques, anti-rootkit tools help safeguard systems from the clandestine actions of rootkits and restore the security of compromised environments.

References:

How Rootkit Revealer works - Rootkit Revealer documentation. (n.d.). https://documentation.help/Rootkit-Revealer/How\_Rootkit\_Revealer\_Works.htm#:~:text=How%20RootkitRevealer%20Works,that%20at%20the%20lowest%20level.

Pfleeger, C. P., Pfleeger, S. L., & Margulies, J. (2018). Security in Computing: 5th Edition. <https://dl.acm.org/citation.cfm?id=2756777>