Name: Ajay Mallasandra Sanath

Operating System – HOMEWORK 1

1)

Isolation/Protection does not allow one app to have access to other app's memory thereby corrupting or crashing the app. And also if there is a crash in App1 it should not affect the entire system. For this we sandbox the applications and create isolation/protection.

Isolation/Protection can be achiebed using Hardware and Software.

Hardware:

Instruction set:

You have 2 types of instruction sets:

Priveleged instruction: These instructions can't be executed by all the process until they have the access rights. These instruction usually include instructions which have the capability to modify the entire system like the I/O

Non Priveleged Instruction: These intructions are like more specific to a process and they do not a capacity to alter the state of the entire system. These include simple Additions and Subtractions.

User Mode/Kernel mode

Usually applications run in user mode and core operating system components run in kernel mode. So all the privileged instructions must be executed in the kernel mode and unprivileged in user mode. When a process running in user mode tries to run privileged intructions it will result in an interrupt and it terminates the process.

A process must make a system call to switch from user mode to kernel mode.

Memory Protection:

The OS kernel must be protected from unauthorized processes trying to run the code. We should not allow a process to access the data structures of kernel and try to corrupt it. We should also see that the data structures of one process will

not to try to access data structuresof another process and corrupt it. Each process has its own stack and it can't try to access other process' stack

Software Features

System Call

system call is how a program requests a service from an operating system's kernel. This may include hardware-related services (for example, accessing a hard disk drive), creation and execution of new processes, and communication with integral kernel services such as process scheduling. System calls provide an essential interface between a process and the operating system

So with all these features it is guaranteed that isolation protection will be enforced.

So in order to execute a privileged instruction the process must be in the kernel mode and then execute instructions which control the entire basic system. And moreover a process can't try to access other process' stack and resources and this guarntess security. This also makes sure that if there is a crash in one process it won't affect the entire system

2)

Read system call - Usually a read system call is used to read or access data from a file. This reads data in bytes from a file and places it in a buffer which is supplied by the calling process.

The following things happen when there is a system call "read"

-The return address of the process will be pushed on the stack.

-There is a number associated with each system call, and read also has a number associated with it. The sytem call interface maintains a table which has all the system calls indexed with their associated numbers. The sytem call interface then makes a mapping with the read() and the table and invokes the read system call.

-There is a switch from user mode to the kernel mode

- Now in the kernel mode a duplicate stack is created. The reason for this is that there is possibility that some hacjker may overwrite the return address on the user stack so that it now runs some malicious code. So in order to prevent the stack is duplicated in the kernel and it knows exactly where to return.

-Now the kernel performs the read task.The result of this operation is stored on the stack and using the return address in the duplicate stack, it is returned to the right place.

Now the process switches from the kernel mode and starts executing in the user mode.

Problem 4:

Multics VS modern OS like Linux

The Differneces:

Modern OS's have static linking at compile time while Multics had dynamic linking. Due to this dynamic linking it was slow

In Multics the files looked like memory while in the modern OS's memory is a separate resource

Similarities:

Multics was the first to use page-segmented storage. Linux and other modern OS's also use the paging and segmentation

The multi user concept- where one computer shared between many users

Ability of one process to create another process

Both have time sharing

Self hosted development

The concept of Isolation/protection is used in both. This provides security from one process to another.

Hierrachial system which has the concept of a root followed by other sub-directories.

3)

Isolation/Protection can be achieved purely with the help of software by sandboxing. A sandbox is a security mechanism for separating running processes. The sandbox typically provides a tightly controlled set of resources for guest programs to run in, such as scratch space on disk and memory. Network access, the ability to inspect the host system or read from input devices are usually disallowed or heavily restricted. In this sense, sandboxes are a specific example of virtualization.

For example, your web browser essentially runs web pages you visit in a sandbox. They're restricted to running in your browser and accessing a limited set of resources — they can't view your webcam without permission or read your computer's local files. If websites you visit weren't sandboxed and isolated from the rest of your system, visiting a malicious website would be as bad as installing a virus.

Pros:

Provides a good security. Any rogue programs will not have access to other process' resources

Cons:

The implementation is not simple. There are some memory overheads

Hardware based implementation – is usually fast compared to software based implemenatation. And provides better performance compared to software

Cons: can be expensive