Module 4: Software Security

CSC3207 Computer Security

Outline

- The Confinement problem
- Virtual machines
- Sandboxes
- Proof carrying code

References

- Chapter 16 Introduction to Computer Security by Matt Bishop

The Confinement Problem

- the problem of confining a program during its execution so that it cannot transmit information to any other program except its caller
- Problem of preventing a server from leaking information that the user of the service considers confidential
- Confinement: ensure application does not deviate from pre-approved behavior
  - ensure misbehaving app cannot harm rest of system
- Confinement is a technique used by an operating system on a suspected program to help ensure that possible damage does not spread to other parts of a system

The Confinement Problem

- A confined program is strictly limited in what system resources it can access
- If a program is not trustworthy, the data it can access are strictly limited
- Strong confinement would be particularly helpful in limiting the spread of viruses
  - Since a virus spreads by means of transitivity and shared data, all the data and programs within a single compartment of a confined program can affect only the data and programs in the same compartment
  - Therefore, the virus can spread only to things in that compartment; it cannot get outside the compartment.

Example Problem

- Server balances bank accounts for clients
- Server security issues:
  - Record correctly who used it
  - Send only balancing info to client
- Client security issues:
  - Log use correctly
  - Do not save or retransmit data client sends

Running Untrusted Code

- We often need to run buggy/unstrusted code
  - programs from untrusted Internet sites - mobile apps, Javascript, browser extensions
  - exposed applications: browser, pdf viewer, outlook
  - legacy daemons: sendmail, bind
  - honeypots

Total Isolation

- Process cannot communicate with any other process and cannot be observed
- Impossible for this process to leak information
- Not practical as process uses observable resources such as CPU, secondary storage,networks, etc.

Confinement

- Confinement is a mechanism for enforcing the principle of least privilege
- A properly confined process cannot transmit data to a second process unless the transmission is needed to complete their task
- The problem is that the confined process needs access to the data to be transmitted and so the confinement must be on the transmission, not on the data access
- To complicate matters, the process may have to transmit some information to the second process
  - In this case, the confinement mechanism must distinguish between transmission of authorized data and transmission of unauthorized data

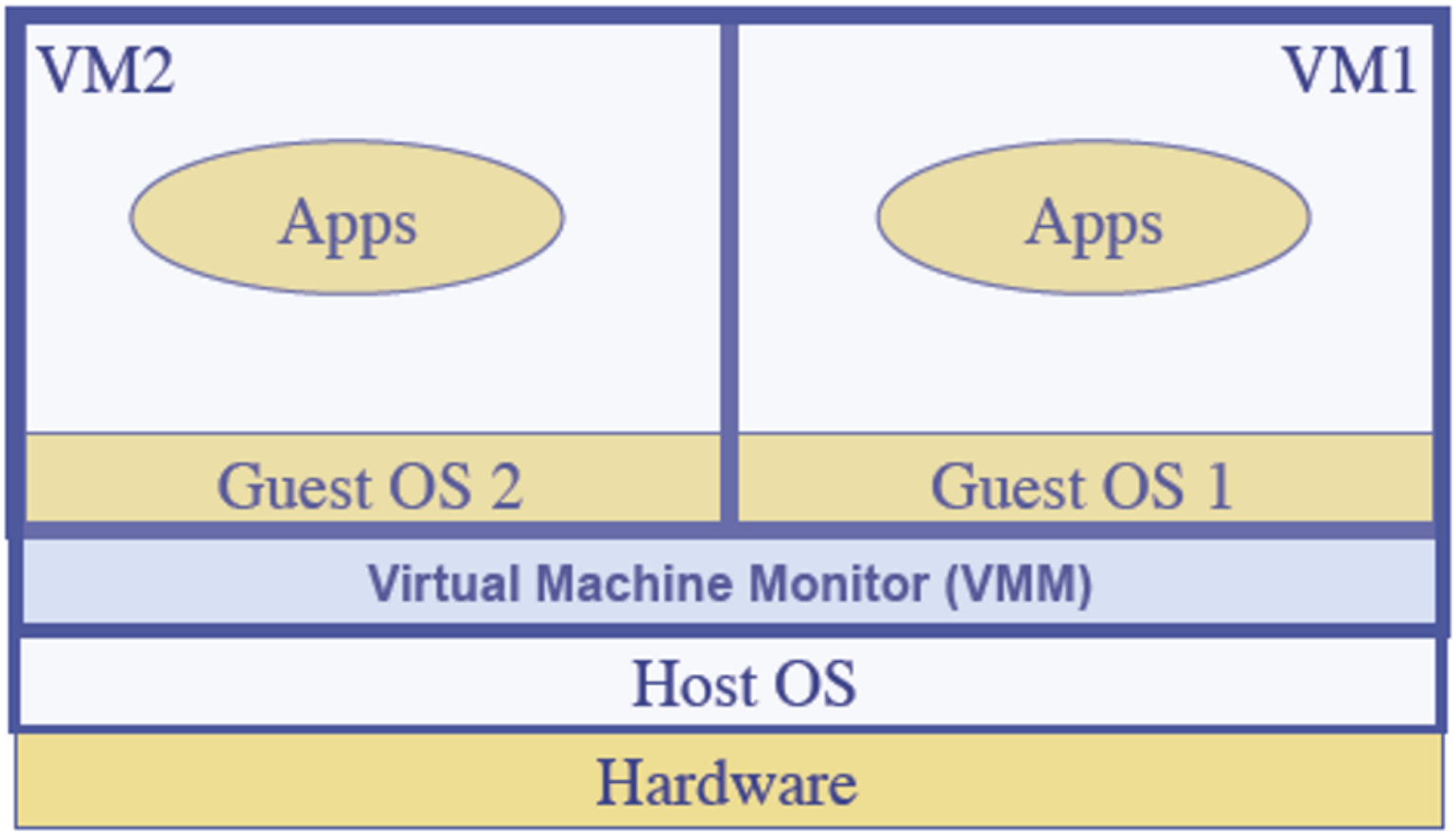
Approaches to Isolation

- Virtual machines
  - Emulate computer
  - Process cannot access underlying computer system and anything not part of that system
- Sandboxing
  - Does not emulate computer
  - Alters interface between computer and process

Virtual Machine

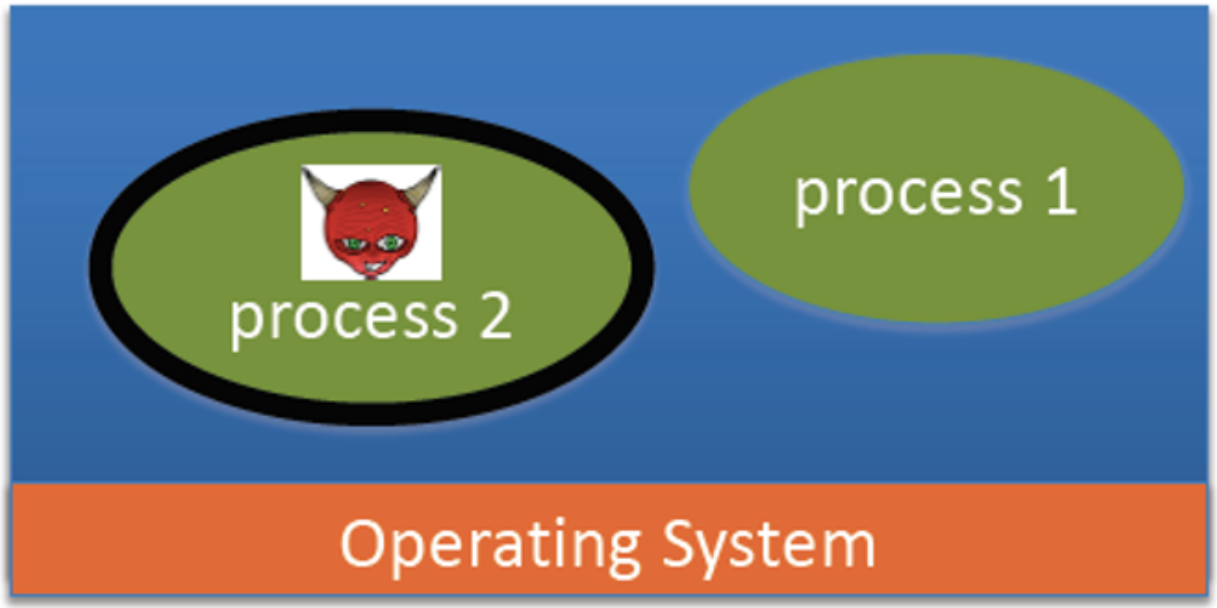
- A program that simulates hardware of computer system
- isolate OS's on a single machine
- existing operating systems do not need to be modified
- Virtual machine monitor (VMM) provides VM on which conventional OS can run (without modifications)
- The virtual machine monitor functions as a security kernel
- Each VM is one subject
- VMM knows nothing about processes running on each VM
- VMM mediates all interactions of VM with resources and other VMS
- An integral component of the VMM is an auditing mechanism which records actions for later analysis

Virtual Machines



Virtual Machines

Isolate a process in a single operating system



Sandboxing

- Sandboxing is a security mechanism and a software technique used to isolate and contain untrusted or potentially harmful code, applications, or processes within a restricted and controlled environment
- Environment in which actions of process are restricted according to security policy
- This isolation prevents the untrusted code from affecting the host system or other applications, enhancing security and minimizing the impact of security vulnerabilities and threats
- Sandboxing is commonly employed in computer systems, web browsers, mobile operating systems, and various other software environments to maintain the integrity and safety of the overall system

Sandbox

- Environment in which actions of process are restricted according to security policy
- Can add extra security-checking mechanisms to libraries, kernel
- Program to be executed is not altered
- Can modify program or process to be executed
- Similar to debuggers, profilers that add breakpoints
- Add code to do extra checks (memory access, etc.) as program runs (software fault isolation)

Sandbox

- The Java virtual machine, in which downloaded applets are executed, is another example of a sandbox
- The sandbox restricts the set of files that the applet can access and the hosts to which the applet can connect

Implementing Confinement

- Key component: reference monitor
- Mediates requests from applications
- Enforces confinement
- Implements a specified protection policy
- Must always be invoked: Every application request must be mediated
- Tamperproof:
  - Reference monitor cannot be killed... or if killed, then monitored process is killed too

Proof-Carrying Code

Proof-Carrying Code

- Downloading software over the network is nowadays common-place
  - But who says that the software does what it promises to do?
  - Who protects the consumer from malicious software or other undesirable side-effects?
- Mechanisms for ensuring that a program is "well-behaved" are needed
- PCC has many uses in systems whose trusted computing base is dynamic, either because of mobile code or because or regular bug fixes or updates
- Examples include extensible operating systems, Internet browsers capable of downloading code, active network nodes and safety-critical embedded controllers

Proof-Carrying Code

- Goal: Safe execution of untrusted code
- PCC is a software mechanism that allows a host system to determine with certainty that it is safe to execute a program supplied by an untrusted source
- Method: Together with the code, a certificate describing its behaviour is sent
  - This certificate is a condensed form of a formal proof of this behaviour
  - Before execution, the consumer can check the behaviour, by running the proof against the program

Discussion Questions

- A program is written to compute the sum of the integers from 1 to 10. The programmer, well trained in reusability and maintainability, writes the program so that it computes the sum of the numbers from k to n. However, a team of security specialists scrutinizes the code. The team certifies that this program properly sets k to 1 and n to 10; therefore, the program is certified as being properly restricted in that it always operates on precisely the range 1 to 10.
  List different ways that this program can be sabotaged so that during execution it computes a different sum, such as 3 to 20
- One way to limit the effect of an untrusted program is confinement:controlling what processes have access to the untrusted program and what access the program has to other processes and data. Explain how confinement would apply to the earlier example of the program that computes the sum of the integers 1 to 10.