Software Security

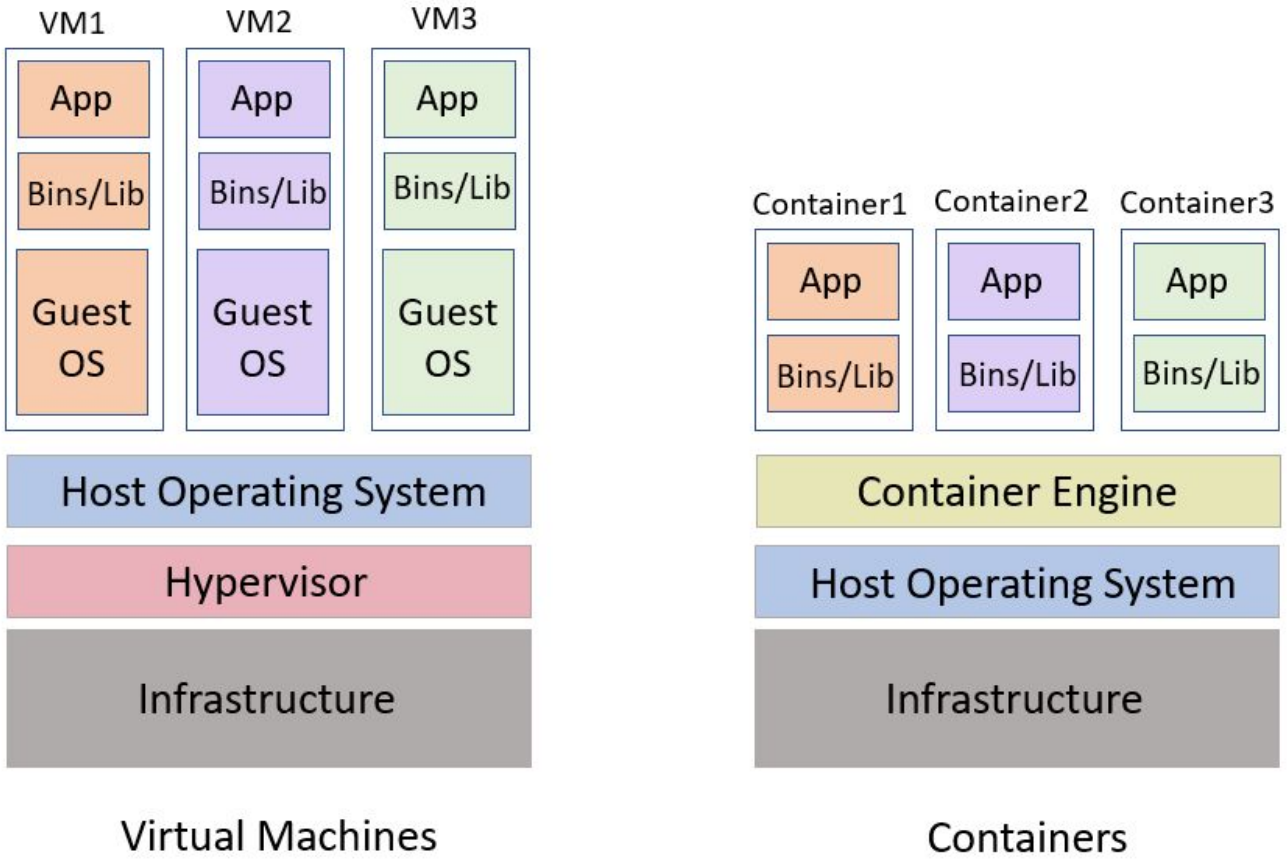
Lecture 2

Isolation

Isolation separates two components from each other and confines their interactions to a well-defined security monitor.

- **Process Abstraction** : Each process has its own virtual memory address space and can interact with other processes only through the operating system which has the role of a security monitor in this case
- **Containers** : The container isolation mechanism separates groups of processes by virtualizing operating system mechanisms such as process identifiers (pids), networking, inter process communication, file system.
- **SFI** : Software-based Fault Isolation (SFI) [33,34] is a software technique to isolate different components in the same address space. Each memory read or write of a component is restricted to the memory area of the component.

Virtual Machine VS Containers



Least Privilege

The principle of least privilege guarantees that a component has the least amount of privileges needed to function.

Scenario :

Sayeed, a junior IT technician, is responsible for **maintaining the company's network**. He is **given access to network management tools and logs** but is not granted access to sensitive HR data or financial records. He **does not have administrative access** to servers that contain sensitive information **unless there's a specific job that requires it**. In such a case, **access is granted temporarily** and revoked once the task is completed. **This approach ensures** that if Sayeed's account is compromised or if he accidentally makes an error, **the damage is limited only to the systems he is permitted** to access, reducing the overall risk to the organization.

Compartmentalization

The idea behind compartmentalization is to break a complex system into small components that follow a well-defined communication protocol to request services from each other.

Scenario :

Consider a compartmentalized payment processing system:

- **User Compartment:** This handles **user authentication and profile management**. Users interact with this compartment *to* **log in and manage their personal information**.
- **Transaction Compartment:** This compartment handles all financial transactions. When a user **initiates a payment, or gets a payment** the transaction is processed here, without exposing sensitive user information from other compartments.
- **Audit Compartment:** An auditing compartment **monitors and logs all transactions**, but it cannot initiate or modify any transaction data.

If the transaction compartment is compromised, the attacker would only gain access to transaction data and not to user credentials or audit logs. This limits the potential damage and makes it easier to isolate and address the security issue.

Compartmentalization VS Isolation

Compartmentalization: Involves logical or functional separation within the same system. Compartments may still interact with each other in a controlled manner.

Isolation: Involves more stringent separation, often preventing any interaction between the isolated components. Systems or processes are kept entirely independent of each other.

Threat Model

Threat modeling is the process of **enumerating and prioritizing** all potential **threats** to a system according to their impact and probability.

Threat modeling evaluates questions such as:

- What are the high value-assets in a system?
- Which components of a system are most vulnerable?
- What are the most relevant threats?

Scenario : Any Login Service

Functionalities :

- The system can authenticate a user based on a username and password through a trusted communication channel.
- Regular users can change their own password.
- Super users can create new users and change any password.

Continued

An incomplete list of possible threats:

- **Implementation flaw** in the authentication service allowing either a user (authenticated or unauthenticated) to authenticate as another user or privileged user without supplying the correct password.
- **Implementation flaw** in privileged user management which allows an unauthenticated or unprivileged user to modify arbitrary data entries in the data storage.
- **Information leakage** of the password from the data storage, allowing an offline password cracker to probe a large amount of passwords.
- **A brute force attack** against the login service can probe different passwords in the bounds of the rate limit.
- The underlying data storage can be compromised through another privileged program overwriting the file or data corruption. Ex : Buffer Overflow

While all vulnerabilities are bugs, not all bugs are vulnerabilities

A **software bug** is therefore a flaw in a computer program that causes it to misbehave in an unintended way. Software bugs are **due to human mistake** in the source code, compiler, or runtime system. Bugs **result in crashes** and unintended program state. Software bugs are **triggered through specific input** (e.g., console input, file input, network input, or environmental input).

If the **bug** can be **controlled by an adversary** to escalate privileges, e.g., gaining code execution **(running shell script which will crash or steal data)**, changing the system state **(making os obsolete)**, or leaking system information **(sending sensitive information to adversaries' server)** then it is called a vulnerability.

A vulnerability requires three key components:

- System is susceptible to bug
- Adversary has access to the bug
- Adversary has capability to exploit the bug.

More examples

Bug: A calculator app incorrectly adding 2 + 2 as 5 is a bug. It affects functionality but doesn't necessarily allow malicious activity.

Vulnerability: A bug in a web application that allows SQL injection is a vulnerability. It can be exploited by attackers to bypass security controls and gain access to sensitive data.

SQL injection

The following command gets executed if user wants to log in.

SELECT * FROM users WHERE username = 'user_input'

If the application doesn’t properly sanitize user input, an attacker could enter the ' OR '1'='1 in the username field

SELECT * FROM users WHERE username = '' OR '1'='1'