

Computer Security: Principles and Practice

Fourth Edition

By: William Stallings and Lawrie Brown

Chapter 12

Operating System Security

Introduction to Operating System Security

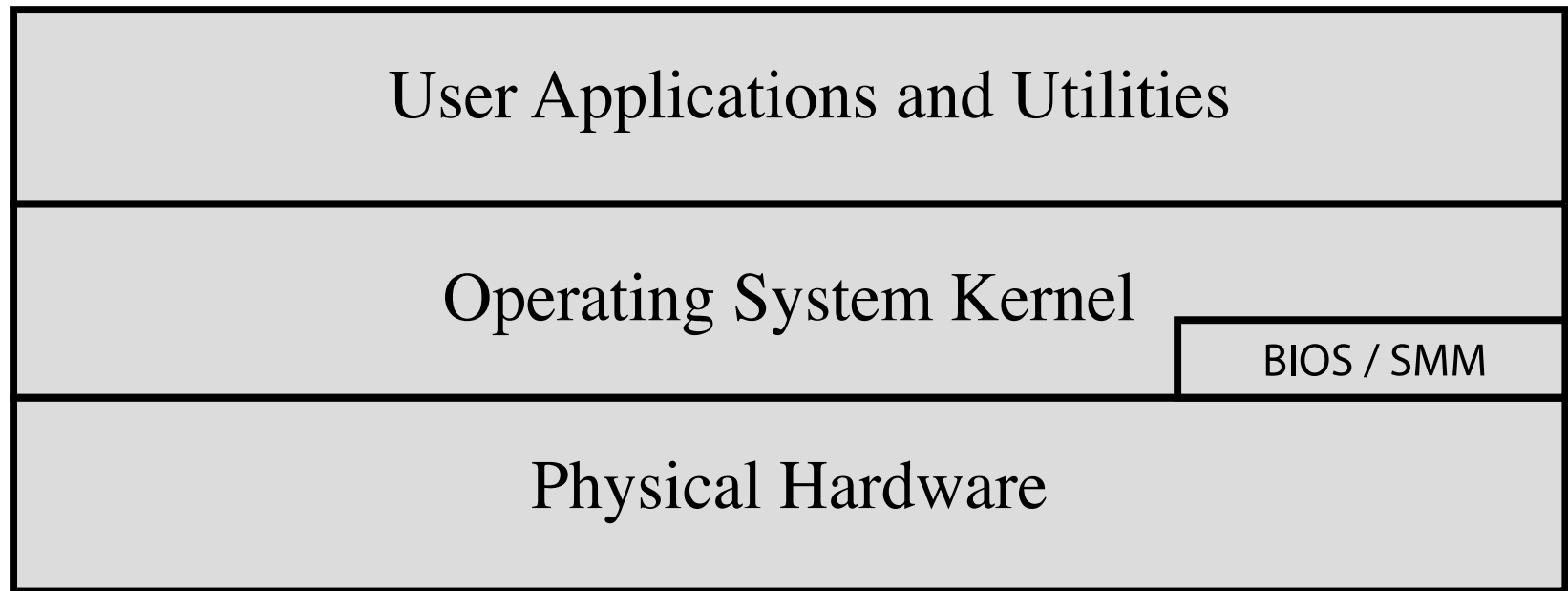


Figure 12.1 Operating System Security Layers

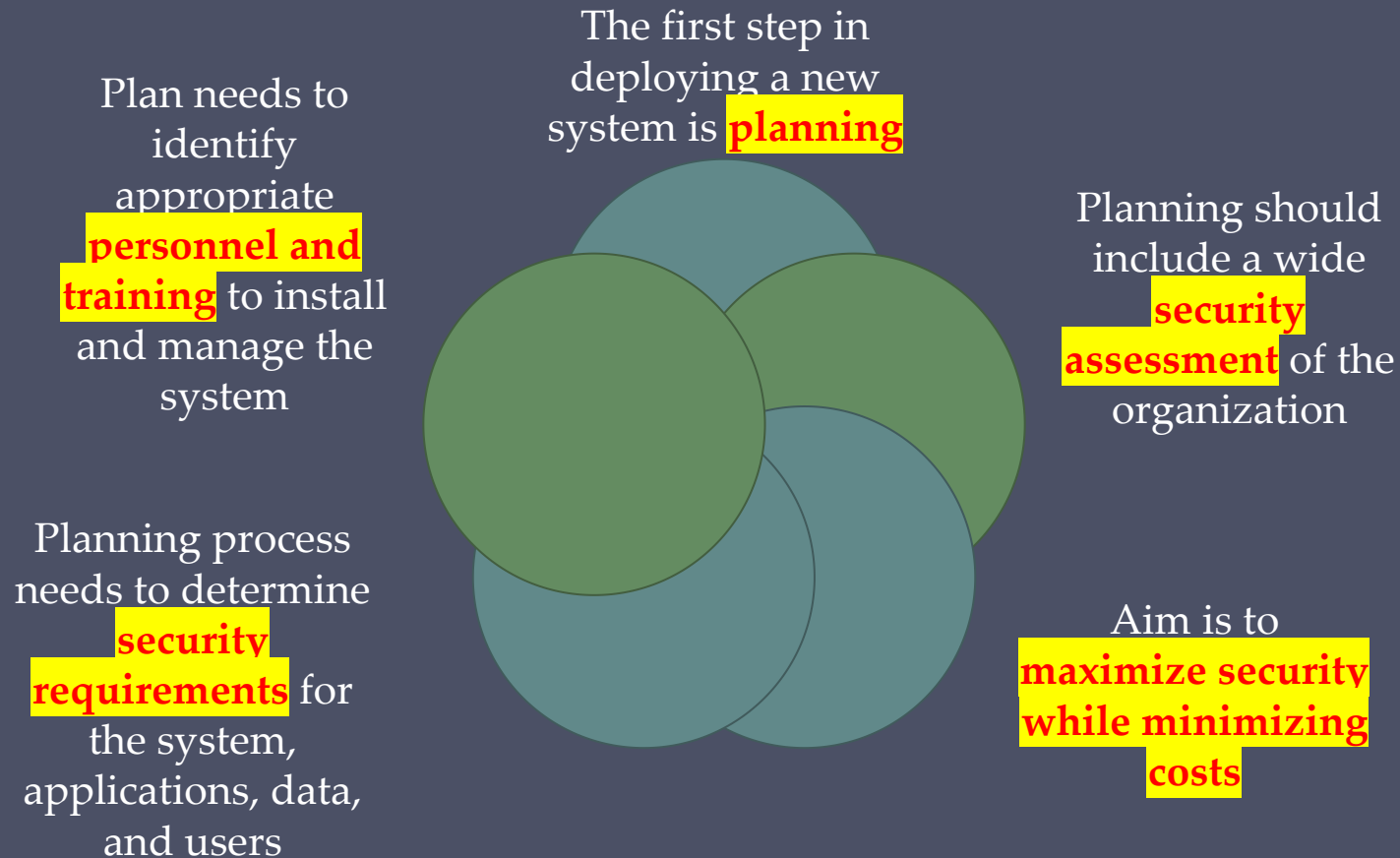
Strategies

- The 2010 Australian Signals Directorate (ASD) lists the “**Top 35 Mitigation Strategies**”
- **Over 85%** of the targeted cyber intrusions investigated by ASD in 2009 could have been prevented
- The **top four strategies** for prevention are:
 - **White-list** approved applications
 - **Patch** third-party applications and operating system **vulnerabilities**
 - **Restrict** administrative **privileges**
 - Create a **defense-in-depth** system
- These strategies largely align with those in the “20 Critical Controls” developed by DHS, NSA, the Department of Energy, SANS, and others in the United States

Operating System Security

- Possible for a system to be **compromised** during the **installation process** before it can install the latest patches
- **Building and deploying** a system should be a **planned** process designed to counter this threat
- Process must:
 - Assess risks and plan the **system deployment**
 - Secure the **underlying operating system** and then the **key applications**
 - Ensure any **critical content** is secured
 - Ensure appropriate **network protection** mechanisms are used
 - Ensure appropriate processes are used to **maintain security**

System Security Planning



System Security Planning Process

The purpose of the system, the type of information stored, the applications and services provided, and their security requirements

Who will administer the system, and how they will manage the system (via local or remote access)

Any **additional security measures** required on the system, including the use of host firewalls, anti-virus or other malware protection mechanisms, and logging

The categories of users of the system, the privileges they have, and the types of information they can access

What access the system has **to information stored on other hosts**, such as file or database servers, and how this is managed

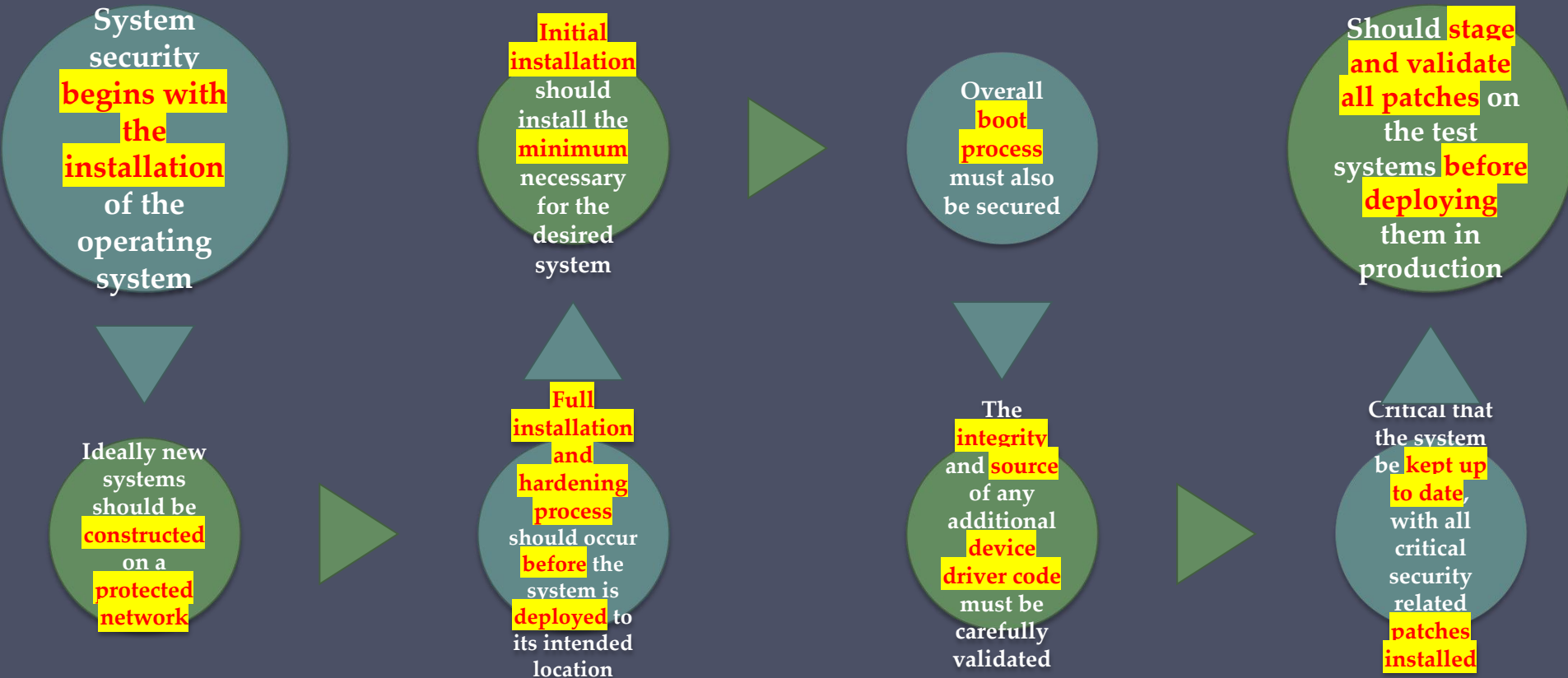
How the users are **authenticated**

How access to the information stored on the system is managed

Operating Systems Hardening

- First critical step in securing a system is to **secure the base operating system**
- **Basic steps**
 - **Install** and **patch** the operating system
 - **Harden** and **configure** the operating system to adequately address the identified security needs of the system by:
 - **Removing unnecessary** services, applications, and protocols
 - **Configuring** users, groups, and **permissions**
 - **Configuring** resource **controls**
 - Install and configure **additional security controls**, such as anti-virus, host-based firewalls, and intrusion detection system (IDS)
 - **Test the security** of the basic operating system to ensure that the steps taken adequately address its security needs

Initial Setup and Patching





Remove Unnecessary

Services,
Applications,
Protocols

- If **fewer software packages** are available to run the **risk is reduced**
- System planning process should **identify what is actually required** for a given system

- When performing the initial installation the **supplied defaults should not be used**
 - Default configuration is set to **maximize ease of use and functionality** rather than security
 - If additional packages are needed later they can be **installed when required**



- **Not all users** with access to a system will **have the same access** to all data and resources on that system
- **Elevated privileges** should be **restricted** to only those users that require them, and then only when they are needed to perform a task

- System **planning** process should consider:
 - **Categories of users** on the system
 - **Privileges** they have
 - **Types of information** they can access
 - How and where they are **defined** and **authenticated**
- **Default accounts** included as part of the system installation should be **secured**
 - Those that are **not required** should be either **removed** or **disabled**
 - Policies that apply to **authentication credentials** configured



- Once the users and groups are defined, appropriate **permissions** can be set on data and resources
- Many of the security hardening guides provide lists of **recommended changes** to the **default access configuration**

- Further security possible by installing and configuring **additional security tools**:
 - **Anti-virus** software
 - Host-based **firewalls**
 - **IDS** or **IPS** software
 - Application **white-listing**



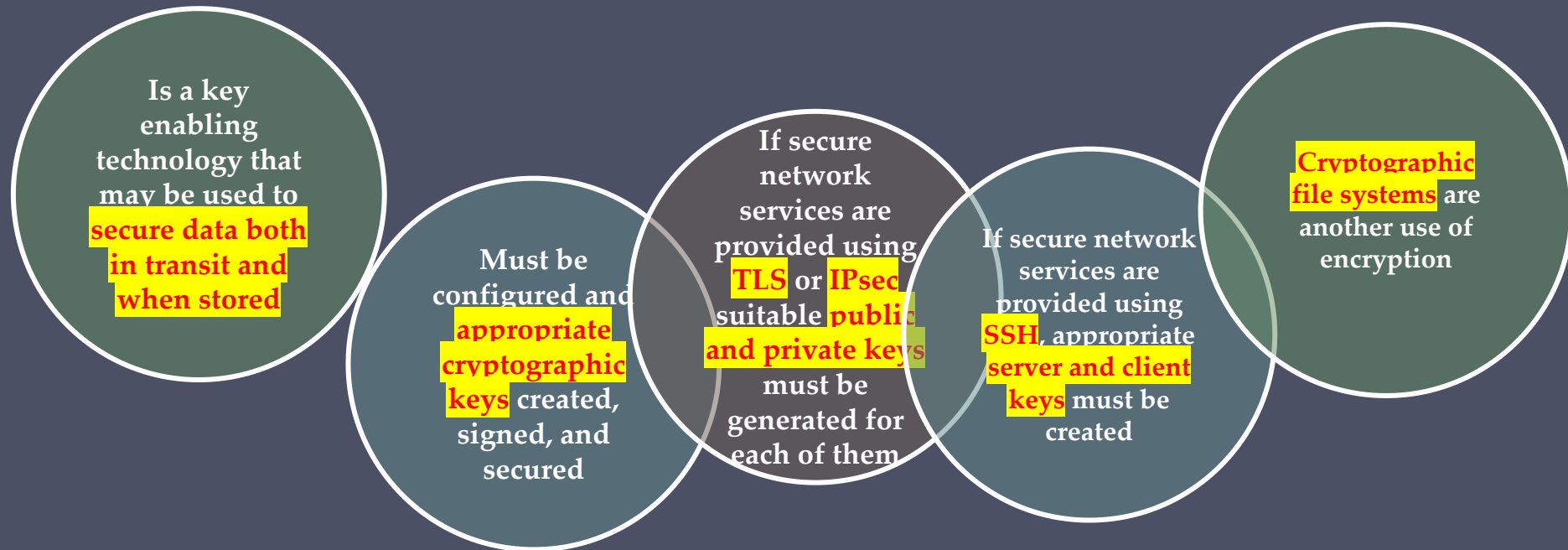
- Final step in the process of initially securing the base operating system is **security testing**
- Goal:
 - Ensure the previous security configuration steps are **correctly implemented**
 - **Identify** any possible **vulnerabilities**

- **Checklists** are included in security hardening guides
- There are **programs** specifically designed to:
 - **Review a system** to ensure that a system **meets the basic security requirements**
 - **Scan for known vulnerabilities** and **poor configuration practices**
- Should be **done following the initial hardening of the system**
- **Repeated periodically** as part of the security maintenance process

Application Configuration

- May include:
 - Creating and specifying appropriate **data storage areas for application**
 - Making **appropriate changes** to the application or service **default configuration details**
- Some applications or services may include:
 - Default data
 - Scripts
 - User accounts
- **Of particular concern with remotely accessed services** such as Web and file transfer services
 - Risk from this form of **attack is reduced** by ensuring that most of the files can **only be read**, but not written, by the server

Encryption Technology



Security Maintenance

- Process of maintaining security is **continuous**
- **Security maintenance** includes:
 - Monitoring and analyzing **logging information**
 - Performing regular **backups**
 - **Recovering** from security compromises
 - Regularly **testing** system security
 - Using appropriate software maintenance processes to **patch** and **update** all critical software, and to **monitor** and **revise** configuration as needed

Logging

Can only **inform you about bad things that have already happened**

In the event of a system **breach** or **failure**, system administrators can more quickly **identify what happened**

Key is to ensure you **capture the correct data** and then appropriately monitor and analyze this data

Information can be **generated by the system, network and applications**

Range of data acquired should be determined during the system **planning** stage

Generates significant volumes of information and it is important that **sufficient space is allocated** for them

Automated analysis is preferred

Data Backup and Archive

Performing regular **backups of data** is a critical control that assists with maintaining the integrity of the system and user data

May be **legal or operational requirements** for the retention of data

Backup

The process of **making copies of data at regular intervals**

Archive

The process of **retaining copies of data over extended periods of time** in order to meet legal and operational requirements to **access past data**

Needs and policy relating to backup and archive should be determined during the system **planning** stage

Kept **online** or **offline**

Stored **locally** or transported to a **remote** site

- Trade-offs include **ease of implementation and cost** versus **greater security and robustness** against different threats

Linux/Unix Security

- **Patch** management
 - **Keeping security patches up to date** is a widely recognized and critical control for maintaining security
- **Application** and **service** configuration
 - Most commonly implemented **using separate text files** for each application and service
 - Generally **located** either in the **/etc directory** or in the **installation tree** for a specific application
 - **Individual user configurations** that can override the system defaults are **located in hidden "dot" files** in each user's home directory
 - Most **important changes** needed to improve system security are to **disable services and applications that are not required**

Linux/Unix Security

- **Users, groups, and permissions**
 - **Access** is specified as granting **read, write, and execute** permissions to each of **owner, group, and others** for each resource
 - Guides **recommend changing** the **access permissions** for **critical directories and files**
 - Local exploit
 - **Software vulnerability** that can be exploited by an attacker to gain **elevated privileges**
 - Remote exploit
 - **Software vulnerability** in a network server that could be **triggered by a remote attacker**

Linux/Unix Security

Remote access controls

- Several **host firewall** programs may be used
- Most systems provide an administrative utility to **select which services will be permitted** to access the system

Logging and log rotation

- Should **not assume** that the **default setting** is necessarily appropriate

Linux/Unix Security

- chroot jail
 - **Restricts the server's view** of the file system to just a **specified portion**
 - Uses **chroot system call** to confine a process by **mapping the root of the filesystem to some other directory**
 - File directories outside the **chroot jail** aren't visible or reachable
 - Main **disadvantage** is **added complexity**

Windows Security

Patch management

- “**Windows Update**” and “**Windows Server Update Service**” assist with regular maintenance and should be used
- **Third party applications** also provide **automatic update** support

Users administration and access controls

- Systems implement **discretionary access controls** resources
- **Vista and later** systems include **mandatory integrity controls**
- **Objects** are labeled as being of **low, medium, high**, or **system integrity level**
- System ensures the **subject's integrity is equal or higher than the object's level**
- Implements a form of the **Biba Integrity model**

Windows Security

Users Administration and Access Controls

Windows systems also define **privileges**

- System wide and granted to user accounts

Combination of share and NTFS permissions may be used to provide **additional security and granularity** when accessing files on a shared resource

User Account Control (UAC)

- Provided in **Vista and later** systems
- Assists with ensuring users with **administrative rights** only **use them when required**, otherwise accesses the system **as a normal user**

Low Privilege Service Accounts

- Used for **long-lived service processes** such as file, print, and DNS services

Windows Security

Application and service configuration

- Much of the configuration information is centralized in the **Registry**
 - Forms **a database of keys and values** that may be queried and interpreted by applications
- **Registry keys** can be directly **modified** using the “**Registry Editor**”
 - More useful for making **bulk changes**

Windows Security

Other security controls

- Essential that **anti-virus**, **anti-spyware**, **personal firewall**, and **other** malware and attack detection and handling software packages are installed and configured
- Current generation Windows systems include **basic firewall** and **malware countermeasure** capabilities
- Important to ensure the set of products in use are **compatible**

Windows systems also support a range of **cryptographic functions**:

- Encrypting files and directories using the **Encrypting File System** (EFS)
- Full-disk encryption with AES using **BitLocker**

“Microsoft Baseline Security Analyzer”

- Free, easy to use tool that **checks for compliance with Microsoft's security recommendations**

Virtualization

- A **technology** that provides **an abstraction of the resources** used by some software which runs in a simulated environment called a **virtual machine (VM)**
- **Benefits** include **better efficiency** in the use of the physical system resources
- Provides support for **multiple distinct operating systems** and associated applications on one physical system
- Raises **additional security concerns**

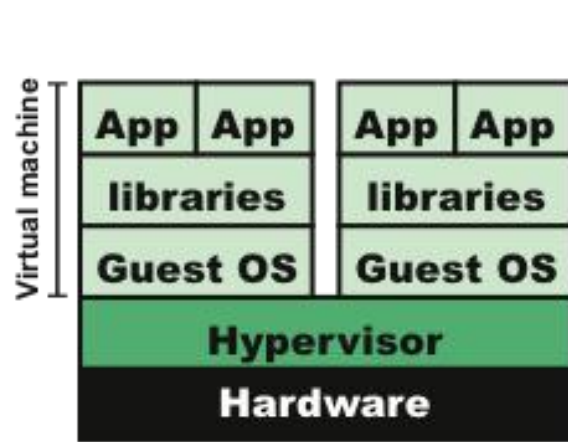
Hypervisor

- **Software** that sits **between the hardware and the VMs**
- Acts as a **resource broker**
- It allows **multiple VMs** to safely **coexist** on a single physical server host and **share** that host's **resources**
- **Virtualizing software** provides **abstraction of all physical resources** and thus enables multiple computing stacks, called **virtual machines**, to be run on a single physical host
- **Each VM** includes an OS, called the **guest OS**
 - This OS may be the same as the host OS, if present, or a different one

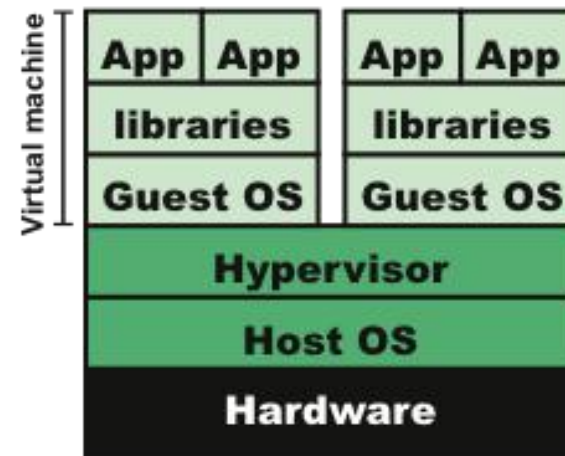
Hypervisor Functions

The **principal functions** performed by a hypervisor are:

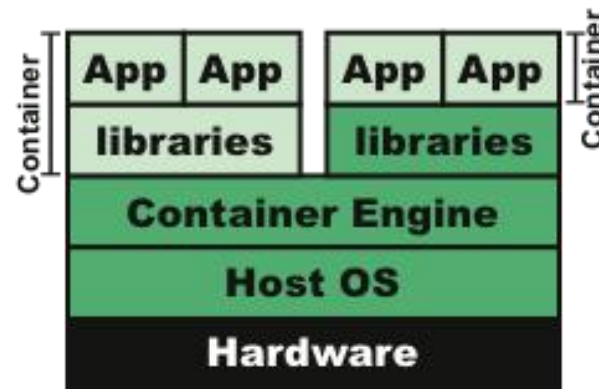
- **Execution management** of VMs
- **Devices emulation** and **access control**
- Execution of **privileged operations** by hypervisor for guest VMs
- **Management of VMs** (also called VM lifecycle management)
- Administration of **hypervisor platform** and **hypervisor software**



(a) Type 1 hypervisor
(native virtualization)



(b) Type 2 hypervisor
(hosted virtualization)



(c) Container (application virtualization)

Figure 12.2 Comparison of Virtual Machines and Containers

Virtualized Systems

- In virtualized systems, the available **hardware resources** must be appropriately **shared among the various guest OS's**
- These include **CPU**, **memory**, **disk**, **network**, and **other attached devices**
- **CPU** and **memory** are generally **partitioned** between these, and **scheduled** as required
- **Disk storage** may be **partitioned**, with each guest having **exclusive use** of some disk resources
- Alternatively, a "**virtual disk**" may be created for each guest, which appears to it as a physical disk with a full file-system, but is viewed externally as a single "disk image" file on the underlying file-system
- Attached devices such as **optical disks**, or **USB devices** are generally **allocated to a single guest OS at a time**

Software Defined Networks (SDNs)

SDNs enable **network segments** to **logically span multiple servers** within and between data centers, while using the same underlying physical network

There are several possible approaches to providing SDNs, including the use of **overlay networks**

- These **abstract all layer 2 and 3 addresses** from the underlying physical network **into whatever logical network structure is required**
- This **structure** can be **easily changed** and **extended** as needed
- The IETF standard **DOVE** (Distributed Overlay Virtual Network) which uses VXLAN (Virtual Extended Local Area Network) can be used to **implement such an overlay network**
- With this flexible structure, it is possible to **locate virtual servers, virtual IDS, and virtual firewalls anywhere within the network as required**

Containers

- A recent approach to virtualization is known as **container virtualization** or **application virtualization**
- In this approach, software known as a **virtualization container**, runs **on top of the host OS kernel** and provides an **isolated execution environment for applications**
- Unlike hypervisor-based VMs, **containers do not aim to emulate physical servers**
- All containerized applications on a host **share a common OS kernel**
- For containers, only **a small container engine** is required as support for the containers
- **Containerization** sits in **between the OS and applications** and incurs lower overhead, but potentially introduces greater security vulnerabilities

Virtualization Security Issues

- **Security concerns** include:
 - Guest OS **isolation**
 - Ensuring that programs **executing within a guest OS** may only access and use the resources allocated to it
 - Guest OS **monitoring by the hypervisor**
 - Which has **privileged access** to the programs and data in each guest OS
 - **Virtualized environment** security
 - Particularly **image and snapshot management** which attackers may attempt to view or modify

Securing Virtualization Systems

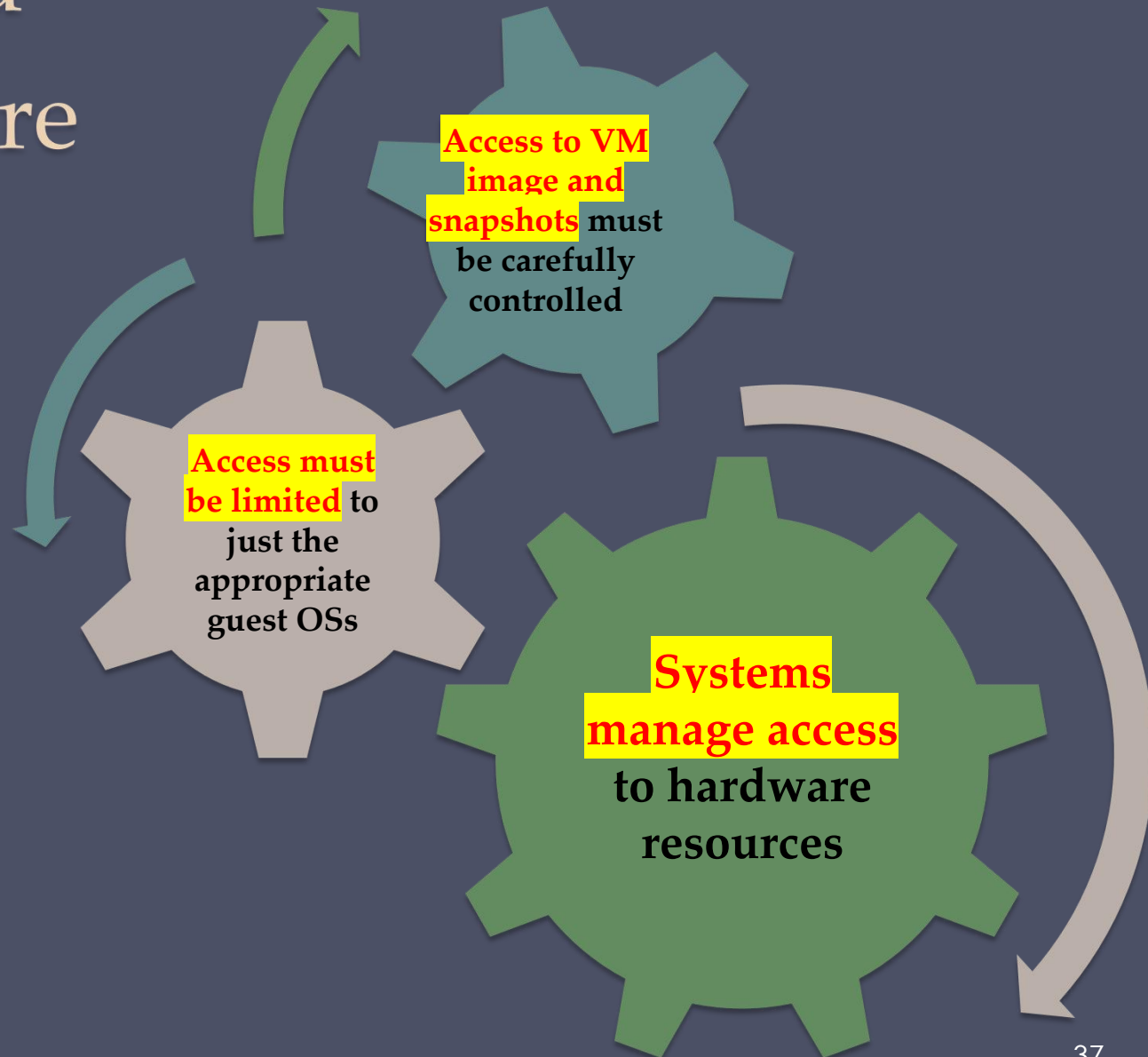
Organizations
using
virtualization
should:

- **Carefully plan** the security of the virtualized system
- **Secure all elements** of a full virtualization solution and maintain their security
- Ensure that **the hypervisor** is properly secured
- Restrict and protect **administrator access** to the virtualization solution

Hypervisor Security

- Should be
 - Secured using a process **similar to securing an operating system**
 - **Installed** in an **isolated environment**
 - **Configured** so that it is **updated automatically**
 - **Monitored** for any **signs of compromise**
 - **Accessed** only by **authorized administration**
- May support **both local and remote administration** so must be configured appropriately
- **Remote administration access** should be considered and secured in the design of any network firewall and IDS capability in use
- **Ideally administration traffic** should use a separate network with very limited access provided from outside the organization

Virtualized Infrastructure Security



Virtual Firewall

Provides firewall capabilities for the network traffic flowing between systems hosted in a **virtualized or cloud environment** that does not require this traffic to be routed out to a physically separate network supporting traditional firewall services

VM Bastion Host

Where a **separate VM** is used as a bastion host supporting the same firewall systems and services that could be configured to run on a physically separate bastion, including possibly IDS and IPS services

VM Host-Based Firewall

Where host-based firewall capabilities provided by the **Guest OS** running on the VM are configured to secure that host in the same manner as used in physically separate systems

Hypervisor Firewall

Where firewall capabilities are provided directly by the **hypervisor**

Summary

- Introduction to operating system security
- System security planning
- Operating systems hardening
 - Operating system installation: initial setup and patching
 - Remove unnecessary services, applications and protocols
 - Configure users, groups, and authentications
 - Configure resource controls
 - Install additional security controls
 - Test the system security
- Application security
 - Application configuration
 - Encryption technology
- Security maintenance
 - Logging
 - Data backup and archive
- Linux/Unix security
 - Patch management
 - Application and service configuration
 - Users, groups, and permissions
 - Remote access controls
 - Logging and log rotation
 - Application security using a chroot jail
 - Security testing
- Windows security
 - Patch management
 - Users administration and access controls
 - Application and service configuration
 - Other security controls
 - Security testing
- Virtualization security
 - Virtualization alternatives
 - Virtualization security issues
 - Securing virtualization systems

作业

- 英文教材（第四版）P443
- Questions 12.3, 12.4, 12.9, 12.11