Introduction to Computer Security

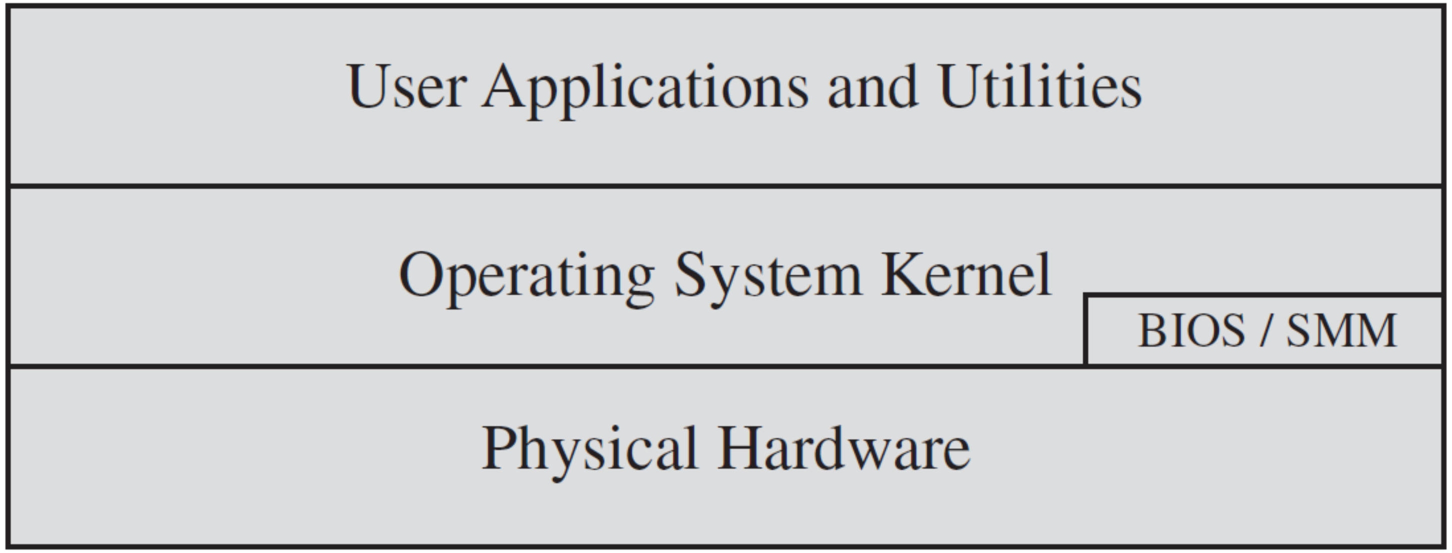
Chapter 12: Operating System Security

Chi-Yu Li (2019 Spring)
Computer Science Department
National Chiao Tung University

Outline

- Introduction
- Operating System Hardening
- Application Security
- Security Maintenance
- Linux/Unix Security
- Windows Security
- Virtualization Security

Operating System Security Layers



User Applications and Utilities

Operating System Kernel

BIOS / SMM

Physical Hardware

- SMM: System Management Mode

Operating System Security

- During the installation process: possible for a system to be compromised
  - □ 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

- The first step in deploying a new system is planning
  - □ A wide security assessment of the organization
  - □ To maximize security while minimizing costs
  - □ To determine security requirements for the system, apps, data, and users
  - □ To identify appropriate personnel and training to install and manage the system

Operating Systems Hardening

- First critical step in securing a system: to secure the base OS

- Basic steps
  - Install and patch the OS
  - Harden and configure the OS to adequately address the identified security needs of the system by
    - Removing unnecessary services, apps, and protocols
    - Configuring users, groups, and permissions
    - Configuring resource controls
  - Install and configure additional security controls
    - E.g., anti-virus, host-based firewalls, and IDS
  - Test the security of the basic OS to ensure that the steps taken adequately address its security needs

Initial Setup and Patching

- Begin with the installation of the OS
- Ideally, new systems should be constructed on a protected network
- Full installation and hardening process should occur before the system is deployed to its intended location
- Initial installation: install the minimum necessary for the desired system
- Overall boot process must also be secured

Initial Setup and Patching (Cont.)

- Integrity and source of any additional device driver code must be carefully validated
- Critical that the system be kept up to date, with all critical security related patches installed
- Should stage and validate all patches on the test systems before deploying them in production

Remove Unnecessary Services, Apps, and Protocols

- If fewer software packages are available to run, the risk is reduced
  - Any of the software packages may contain software vulnerabilities
  - System planning process should identify what is actually required
- When performing the initial installation, the supplied defaults should not be used
  - Why?
  - Default: maximize ease of use and functionality, rather than security
  - Customized installation: only the required packages are installed

Configure Users, Groups, and Authentication

- 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
  - Categories of users on the system
  - Privileges they have
  - Types of information they can access
  - How and where they are defined and authenticated

Configure Users, Groups, and Authentication (Cont.)

- 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 shall be configured
    - E.g., password length, complexity, etc.

Configure Resource Controls

- 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

Install Additional Security Controls

- Further security possible by installing and configuring additional security tools
  - □ Anti-virus software (multi-vendor)
  - □ Host-based firewalls
  - □ IDS or IPS software
  - □ App white-listing

Test the System Security

- Final step in the process of initially securing the base OS: security testing

- Goal:
  - □ Ensure the previous security configuration steps are correctly implemented
  - □ Identify any possible vulnerabilities

- Should be done following the initial hardening of the system
- Repeated periodically as part of the security maintenance process

Application Security

- Application Configuration
  - May include
    - Creating and specifying appropriate data storage areas for app
    - Making appropriate changes to the app or service default configuration details
  - Some apps or services may include
    - Default data
    - Scripts
    - User accounts
  - Of particular concern with remotely access services
    - Web and file transfer services
    - Risk can be reduced: ensuring that most of the files can only be read, but not written

Application Security (Cont.)

- Encryption technology
  - A key enabling technology: used to secure data both in transit and when stored
  - Must be configured and appropriate cryptographic keys created, signed, and secured
  - If secure network services are provided
    - Using TLS or IPSec, suitable public and private keys must be generated for each of them
    - Using SSH, appropriate server and client keys must be created
  - Cryptographic file systems are another use of encryption

Security Maintenance

- Process of maintaining security is continuous

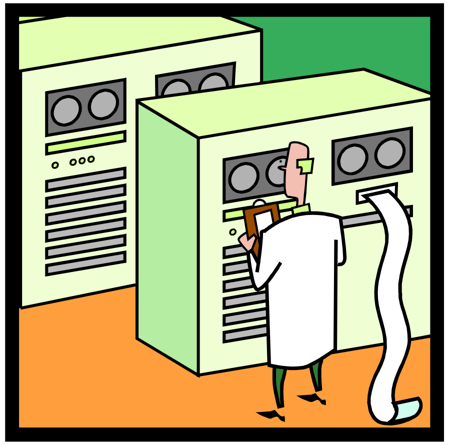
- NIST SP 800-123 suggests to include
  - □ Monitoring and analyzing logging information
  - □ Performing regular backups
  - □ Recovering from security compromises
  - □ Regularly testing system security
  - □ Using appropriate software maintenance processes to patch and update all critical software, and to monitor and revise configuration as needed

Logging

- NIST SP 800-123: Logging is a cornerstone of a sound security posture
  - Key is to ensure you capture correct data and then appropriately monitor and analyze this data
    - Information can be generated by the system, network, and apps
    - Range of data acquired should be determined during the system planning stage
    - Sufficient space is required for significant volumes of information
    - Automated analysis is preferred
  - Can only inform you about bad things that have already happened
  - Can allow administrators to identify what happened

Data Backup and Archive

- Performing regular backups of data is a critical control
  - □ Maintaining the integrity of the system and user 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 to meet legal and operational requirements to access past data



- Needs and policies should be determined during the system planning stage
  - □ Key decisions: online or offline, stored locally or transported to a remote site

Outline

- Introduction
- Operating System Hardening
- Application Security
- Security Maintenance
- Linux/Unix Security
- Windows Security
- Virtualization Security

Linux/Unix Security

- Patch management
  - Keeping security patches up to date is a widely recognized
  - e.g., Red Hat, Fedora: *up2date* or *yum*; Ubuntu, Debian: *apt-get*

- App and service configuration
  - Most commonly implemented using separate text files for each app and service
  - Generally located either in the */etc* directory or in the installation tree for a specific app
    - Individual user configurations can override the system defaults: in each user's home directory
    - Assign proper permission to access them
  - Most important: disabling services and apps that are not required
    - Especially for remotely accessible services

Linux/Unix Security (Cont.)

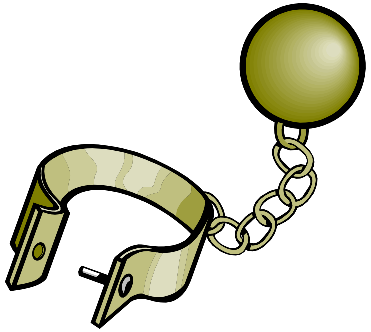
- Users, groups, and permissions
  - □ Discretionary access control
  - □ Access is specified as granting read, write, and execute permissions to each of owner, group, and others for each resource
    - Set by the *chmod* command
    - Extended access rights: *getfacl* and *setfacl* commands
    - Information on user accounts and group membership: stored in the */etc/passwd* and */etc/group* files
  - □ Guides recommend changing the access permissions for critical directories and files
    - Key targets for attackers: programs that set user to root or set group to a privileged group
    - Widely accepted: number and size of setuid root programs should be minimized
  - □ Software vulnerability: local exploit and remote exploit

Linux/Unix Security (Cont.)

- Remote access controls
  - Host firewall programs
    - e.g., using *iptables* to configure the *netfilter* kernel module
  - Network access control mechanisms
    - e.g., TCP Wrappers library and *tcpd* daemon
    - Can use the same policy files: */etc/hosts.allow* and */etc/hosts.deny*
- Logging and log rotation
  - Most apps can be configured to log which levels of detail: debugging to none
    - Either a dedicated file to write app event data, or a syslog facility to use (*/dev/log*)
  - *logrotate* can be configured to rotate any logs on the system
    - Work for *syslogd*, *Syslog-NG*, or individual apps

Linux/Unix Security (Cont.)

- App security using a chroot jail
  - Some network accessible services: do not require access to the full file-system, but rather only need a limited set of data files and directories
    - e.g., FTP
  - Running such services in a chroot jail: restricting the server’s view of the file system to just a specified portion
    - Using the *chroot* system call
    - e.g., mapping the root */* to some other directory */srv/ftp/public*
  - Drawback: added complexity
    - Troubleshooting a chrooted app can be difficult



Linux/Unix Security (Cont.)

- Security testing
  - May follow the system hardening guides provided by the NSA

- Some commercial and open-source tools for security scanning and vulnerability testing
  - Nessus: network vulnerability scanner
    - Originally an open-source tool, commercialized in 2005
  - Tripwire: file integrity checking tool
    - Originally an open-source tool, commercialized later
  - Nmap: network vulnerability scanner
    - Open-source, freeware

Windows Security

- Patch management
  - “Windows Update” and “Windows Server Update Services”
  - Many third-party apps provide automatic update support
- Users administration and access controls
  - Users and groups in Windows systems are defined with a Security ID (SID)
    - Stored in the Security Account Manager (SAM)
    - Information supplied by a central Active Directory (AD) using the LADP protocol
  - Discretionary access controls
  - Vista and letter systems include mandatory integrity controls
  - Privileges to user accounts in User Account Control (UAC): backup the computer, change the system time, modifying system configuration, etc.
  - File access: a combination of share and NTFS permissions

Windows Security (Cont.)

- Application and service configuration
  - Much of the configuration information is centralized in the Registry
    - Forming a database of keys and values that may be queried and interpreted by apps
  - Registry keys can be directly modified using the “Registry Editor”
    - More useful for making bulk changes
- Other security controls
  - Essential: anti-virus, anti-spyware, personal firewall, etc.
    - Important to ensure the set of products in use are compatible
  - Current generation Windows: including some basic firewall and malware countermeasure capabilities
  - Encrypting File system (EFS): encrypting files and directories
  - BitLocker: full-disk encryption with AES

Windows Security (Cont.)

- Security testing
  - "Microsoft Baseline Security Analyzer": a simple, free, easy-to-use tool
    - Helping small- to medium-sized business improve the security of their systems
    - Checking for compliance with Microsoft's security recommendations

Virtualization

- A technology that provides an abstraction of the computing resources used by some software
  - □ Running in a simulated environment called a virtual machine (VM)
- Benefits
  - □ Better efficiency in the use of the physical system resources
  - □ Support for multiple distinct OS and associated apps on one physical system
- However, it raises additional security concerns

Hypervisor

- The software that sits between the hardware and the VMs
  - Acting as a resource broker

- Allowing multiple VMs to safely coexist on a single physical server and share the server's resources

- Providing abstraction of all physical resources, such as processor, memory, network, and storage

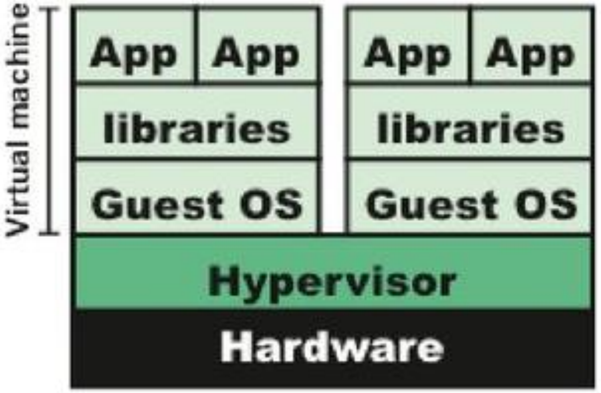
- Host OS v.s. Guest OS on each VM

Principal Functions for Hypervisor

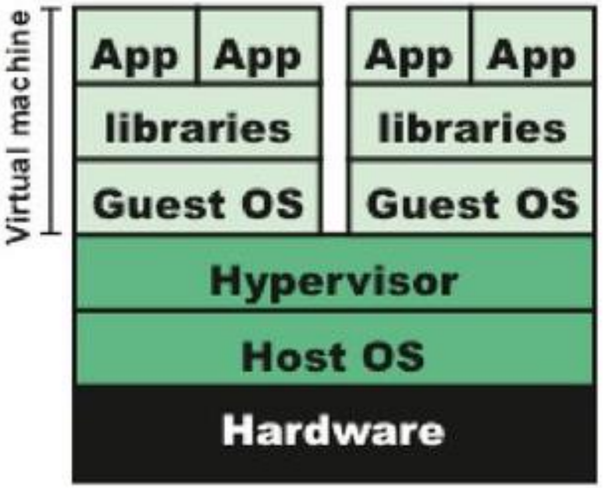
- Execution management of VMs
- Devices emulation and access control
- Execution of privileged operations by hypervisor for guest VMs
- Management of VMs
- Administration of hypervisor platform and hypervisor software

Two Types of Hypervisors

- Distinguished by whether there there is an OS between the hypervisor and the host
    - □ Type 1: native virtualization
        - ■ Hypervisor can directly control the physical resources of the host
    - □ Type 2: hosted virtualization
        - ■ Hypervisor exploits the resources and functions of a host OS



(a) Type 1 hypervisor
(native virtualization)



(b) Type 2 hypervisor
(hosted virtualization)

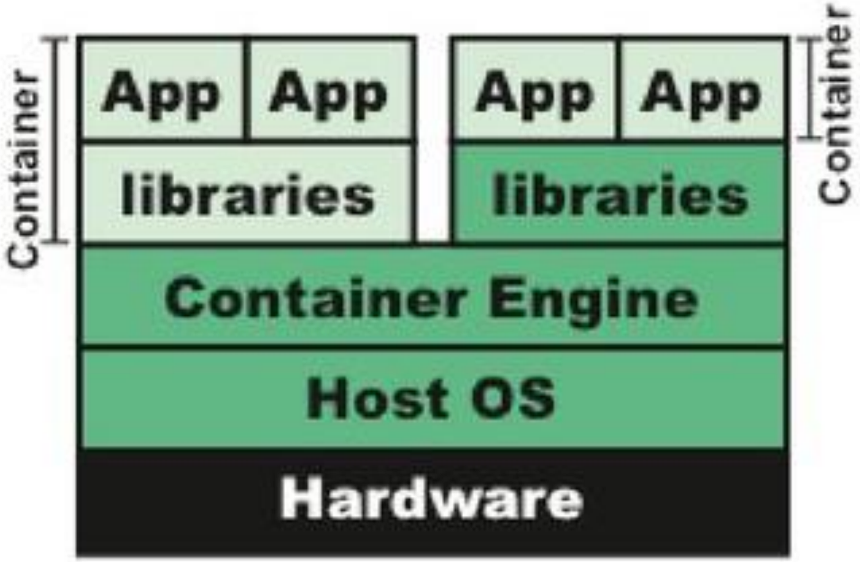
Two Types of Hypervisors: Key Differences

- Typically, type 1 hypervisors perform better
  - Doesn’t compete for resources with an OS
  - More virtual machines can be hosted
- Type 1 hypervisors are more secure
  - VMs on a Type 1 hypervisor cannot affect other VMs or the hypervisor
  - Type 2 hypervisor: a malicious guest could potentially affect more than itself
- Type 2 hypervisors: enabling virtualization without needing to dedicate a server to that function

- Type 1: typically seen in servers; Type 2: more common in clients

New Type: Container/App Virtualization

- Virtualization container runs on top of the host OS kernel and provides an isolated execution environment for apps
  - □ Does not aim to emulate physical servers
  - □ All containerized apps on a host share a common OS kernel
  - □ Each container as an isolated instance
- Reducing overhead: no need of resources to run a separate OS for each app
- But, greater security vulnerabilities



(c) Container (application virtualization)

Virtualization Security Issues

- NIST SP 800-125 (Guide to Security for Full Virtualization Technologies, Jan. 2011)
  - Guest OS isolation: ensure that programs executing within a guest OS may only access and use the resources allocated to it
    - Not covertly interact with programs or data either in other guest OS or in the hypervisor
  - Guest OS monitoring: with privileged access to the programs and data in each guest OS
    - Must be trusted as secure from subversion and compromised use of this access
  - Virtualized environment security
    - Particularly image and snapshot management

Securing Virtualization Systems

- NIST SP 800-125: organizations using virtualization should
  - □ Carefully plan the security of the virtualized system
  - □ Secure all elements of a full virtualization solution
    - Including the hypervisor, guest OS, and virtualized infrastructure
  - □ Ensure that the hypervisor is properly secure
  - □ Restrict and protect administrator access to the virtualization solution

Securing Virtualization Systems (Cont.)

- Hypervisor security
  - Secured using a process similar to securing an OS
  - Installed in an isolated environment
  - Configured so that it is updated automatically
  - Monitored for any signs of compromise
  - Accessed only by authorized administration
    - Both local and remote

Virtualized Infrastructure Security

- Access to hardware resources (e.g., disk, network)
  - Limited to just the appropriate guest OSs that use any resource
- Access to VM images and snapshots
  - Must be carefully controlled
- Traffic should be suitably isolated and protected
  - Management traffic: for hypervisor administration and configuration
  - Infrastructure traffic: for migration of VM images, or connections to network storage
  - App traffic: between apps running VMs and to external networks
    - May be further separated into many segments

Questions?