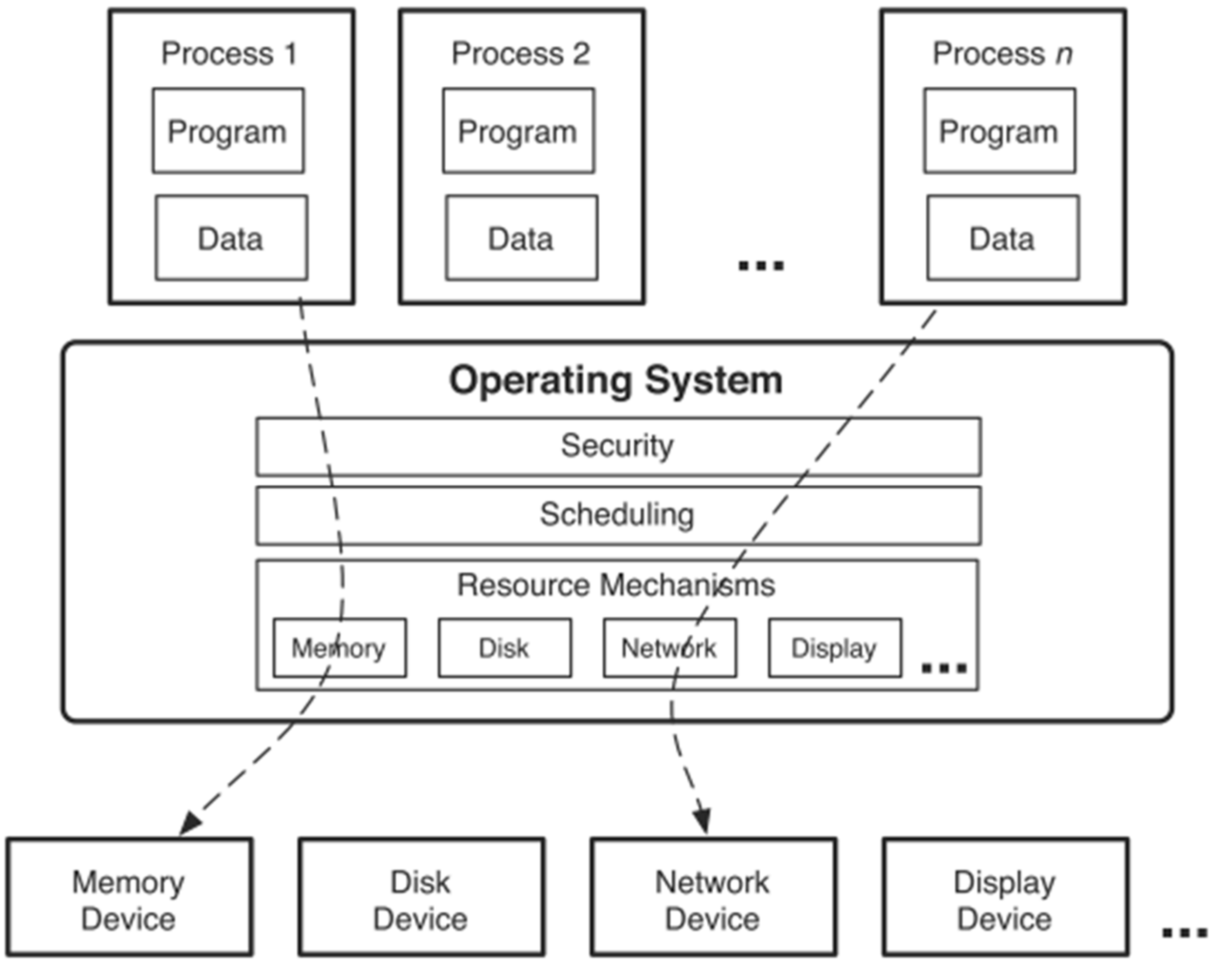
Securing Operating System

Overview



Process 1
Program
Data
Process 2
Program
Data
...
Process n
Program
Data
Operating System
Security
Scheduling
Resource Mechanisms
Memory
Disk
Network
Display
...
Memory Device
Disk Device
Network Device
Display Device
...

Security goals

- Secrecy: limit read
- Integrity: limit write
- Availability: limit consume

system access → which subjects (e.g., processes and users) can perform which operations (e.g., read and write) on which objects (e.g., files and sockets)

limits operations based on a security requirement,

TRUST MODEL

- A ***trust model*** defines a set of software or data upon which system depends for correct enforcement of system security goals
- Trust model of operating system is Trusted Computing Base (TCB)
- A system TCB should consist of the minimal amount of software necessary to enforce the security goals correctly

THREAT MODEL

- A ***threat model*** defines a set of operations that an attacker may use to compromise a system
- The model assume a powerful attacker who is capable of injecting operations from the network and may be in control of some of the running software on the system
- Assume that the attacker may try any and all operations that are permitted to the attacker

ACCESS CONTROL FUNDAMENTALS

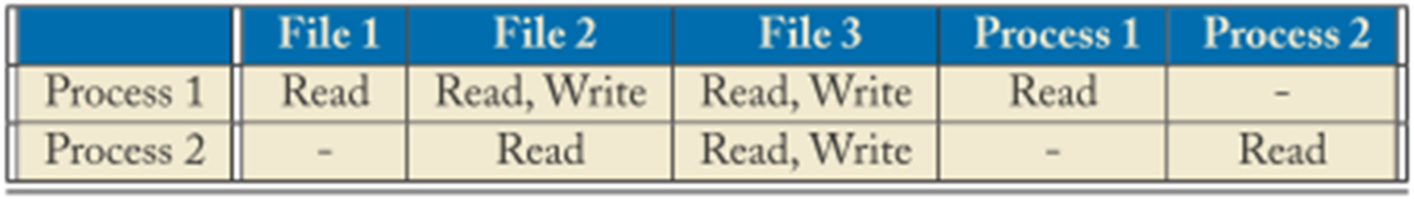
- Fundamental concepts of access control:
  - **Protection system**: defines the access control specification
  - **Reference monitor**: enforces specification

PROTECTION SYSTEM

- A ***protection system*** consists of a ***protection state***, which describes:
  - the operations that system subjects can perform on system objects, and
  - a set of protection state operations, which enable modification of that state.

PS: LAMPSON'S ACCESS MATRIX

- Lampson defined the idea that a protection state is represented by an ***access matrix***



	File 1	File 2	File 3	Process 1	Process 2
Process 1	Read	Read, Write	Read, Write	Read	-
Process 2	-	Read	Read, Write	-	Read

- *access control list (ACL): store columns*
- *C-List (capability list): store rows*

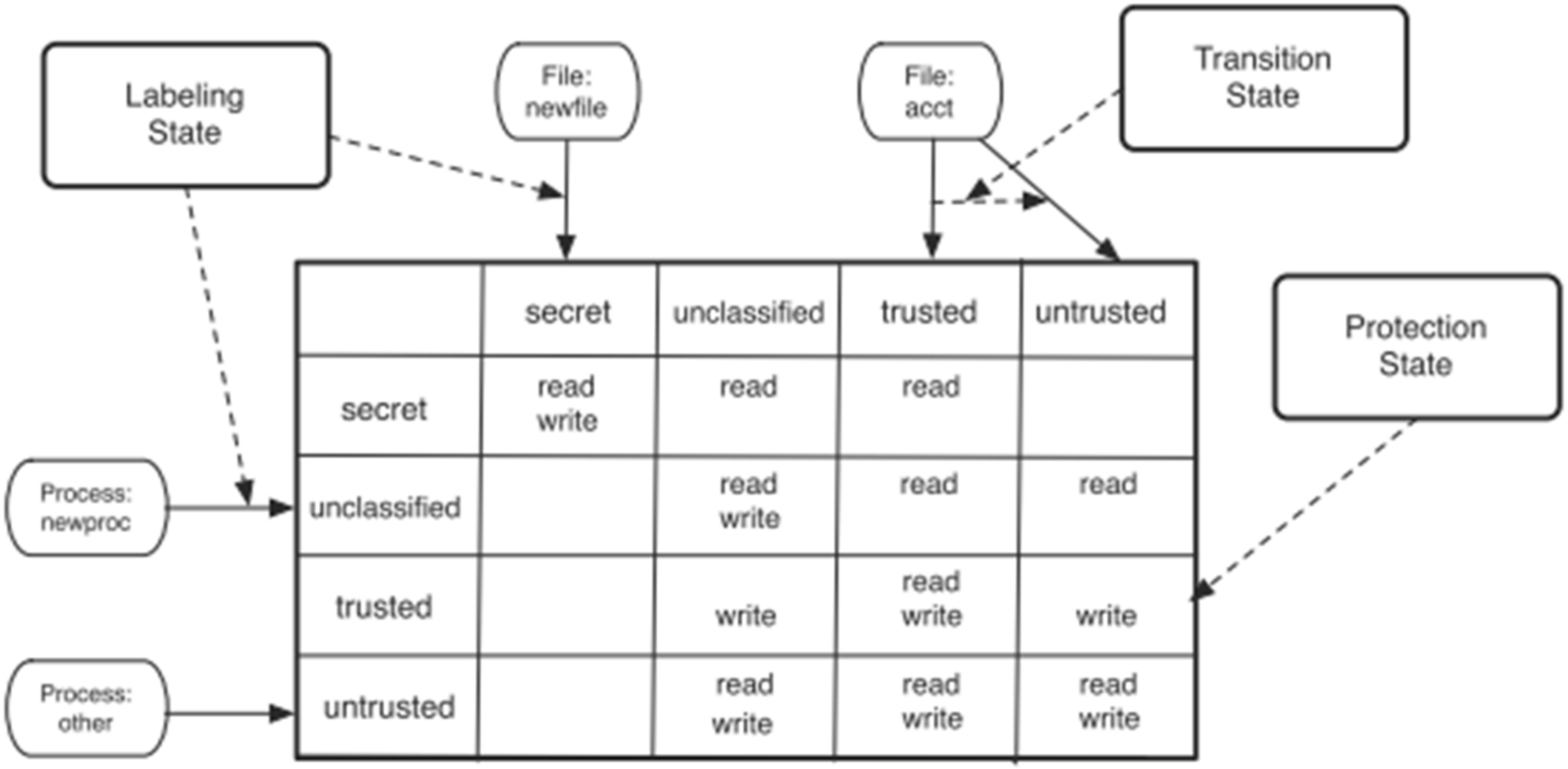
PS: Problem of Access Matrix

- A problem of access matrix: untrusted processes can tamper with the ***protection system***
- Untrusted user processes can add new subjects, objects, or operations assigned to cells
- Discretionary access control (DAC) system: permits untrusted process to modify protection state

PS: MANDATORY PROTECTION SYSTEMS

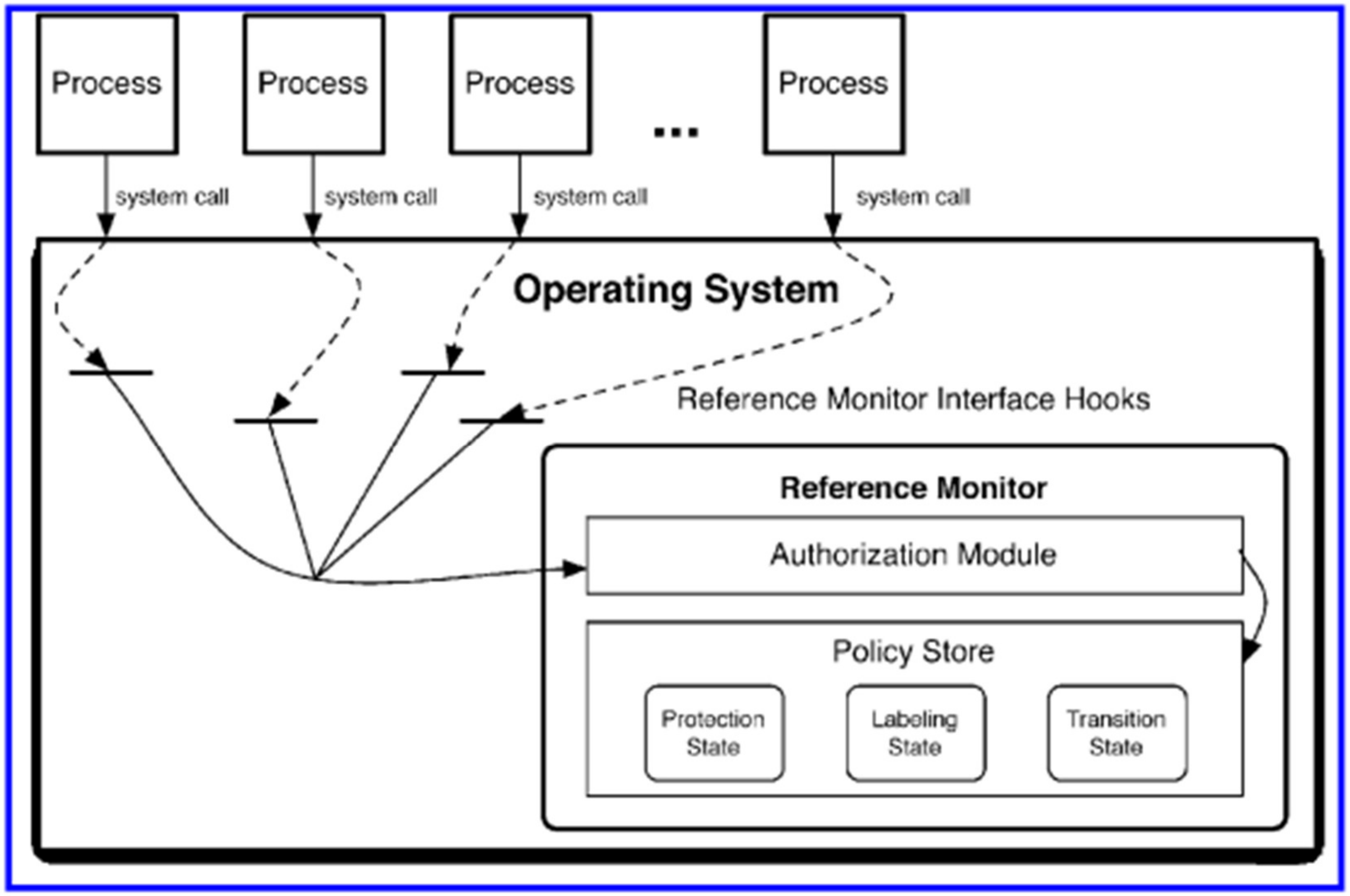
A mandatory protection system is a protection system that can only be modified by trusted administrators via trusted software, consisting of the following state representations:

- ***A mandatory protection state*** is a protection state where subjects and objects are represented by labels where the state describes the operations that subject labels may take upon object labels;
- ***A labeling state*** for mapping processes and system resource objects to labels;
- ***A transition state*** that describes the legal ways that processes and system resource objects may be relabeled.



	secret	unclassified	trusted	untrusted
secret	read write	read	read	
unclassified		read write	read	read
trusted		write	read write	write
untrusted		read write	read write	read write

REFERENCE MONITOR



Reference Monitor Interface

- Defines where the authorization module needs to be invoked to perform an authorization query to the protection state:
  - a labeling query to the labeling state, or
  - a transition query to the transition state
- It ensures that all security-sensitive operations are authorized by the access enforcement mechanism

Authorization Module

- The authorization module determines the exact queries that are to be made to the policy store
- Takes interface's inputs (e.g., process identity, object references, and system call name), and converts these to a query
- The challenge of the module:
  - to map the process identity to a subject label
  - to map the object references to an object label,
  - to determine the actual operations to authorize

Policy Store

- The authorization module is answered by the policy store
- Database for the protection state, labeling state, and transition state
- The policy store responds to authorization, labeling, and transition queries based on the protection system that it maintains

SECURE OPERATING SYSTEM DEFINITION

A secure operating system is an operating system where its access enforcement satisfies the reference monitor concept

The reference monitor concept defines the necessary and sufficient properties of any system that securely enforces a mandatory protection system, consisting of three guarantees:

1. Complete Mediation: The system ensures that its access enforcement mechanism mediates all security-sensitive operations.
2. Tamperproof: The system ensures that its access enforcement mechanism, including its protection system, cannot be modified by untrusted
3. Verifiable:The access enforcement mechanism, including its protection system,“must be small enough to be subject to analysis and tests, the completeness of which can be assured”

Security Kernels

UNIX PROTECTION SYSTEM

- UNIX is a discretionary access control (DAC) system
- UNIX protection system consists of a protection state and a set of operations that enable processes to modify that state
- All UNIX resources are represented as files. The protection state specifies that subjects may perform read, write, and execute operations on files.
- UNIX process identity consists of a user id (UID), a group id (GID)
- Files are also associated with an owner UID and an owner GID.
- A process with the owner UID can modify any aspect of the protection state for this file

UNIX mode bits

- A compressed access control list format
- To specify the access rights of identities to files
- Mode bits define the rights of three types of subjects: (1) the file owner UID; (2) the file group GID;and (3) all other subjects

Using mode bits authorization

- First, the UNIX authorization mechanism checks whether the process identity's UID corresponds to the owner UID of the file, and if so, uses the mode bits for the owner to authorize access.
- If the process identity's GID or supplementary groups correspond to the file's group GID, then the mode bits for the group permissions are used.
- Otherwise, the permissions assigned to all others are used.

Example

UNIX mode bits are of the form {owner bits, group bits, others bits} where each element in the tuple consists of a read bit, a write bit, and an execute bit.

Example the mode bits: rwxr--r--

Name	Owner	Group	Mode Bits
foo	alice	faculty	rwxr--r--
bar	bob	students	rw-rw-r--
baz	charlie	faculty	rwxrwxrwx

UNIX AUTHORIZATION

- UNIX authorization mechanism:
  - controls each process's access to files
  - implements protection domain transitions (enable a process to change its identity)
- The authorization mechanism runs in the kernel
- UNIX authorization mechanism does not implement a reference monitor

UNIX VULNERABILITIES

- Network-facing Daemons
- Rootkits
- Environment Variables
- Shared Resources
- Time-of-Check-to-Time-of-Use(TOCTTOU): untrusted processes may change the state of the system between the time an operation is authorized and the time that the operation is performed.

WINDOWS PROTECTION SYSTEM(1/3)

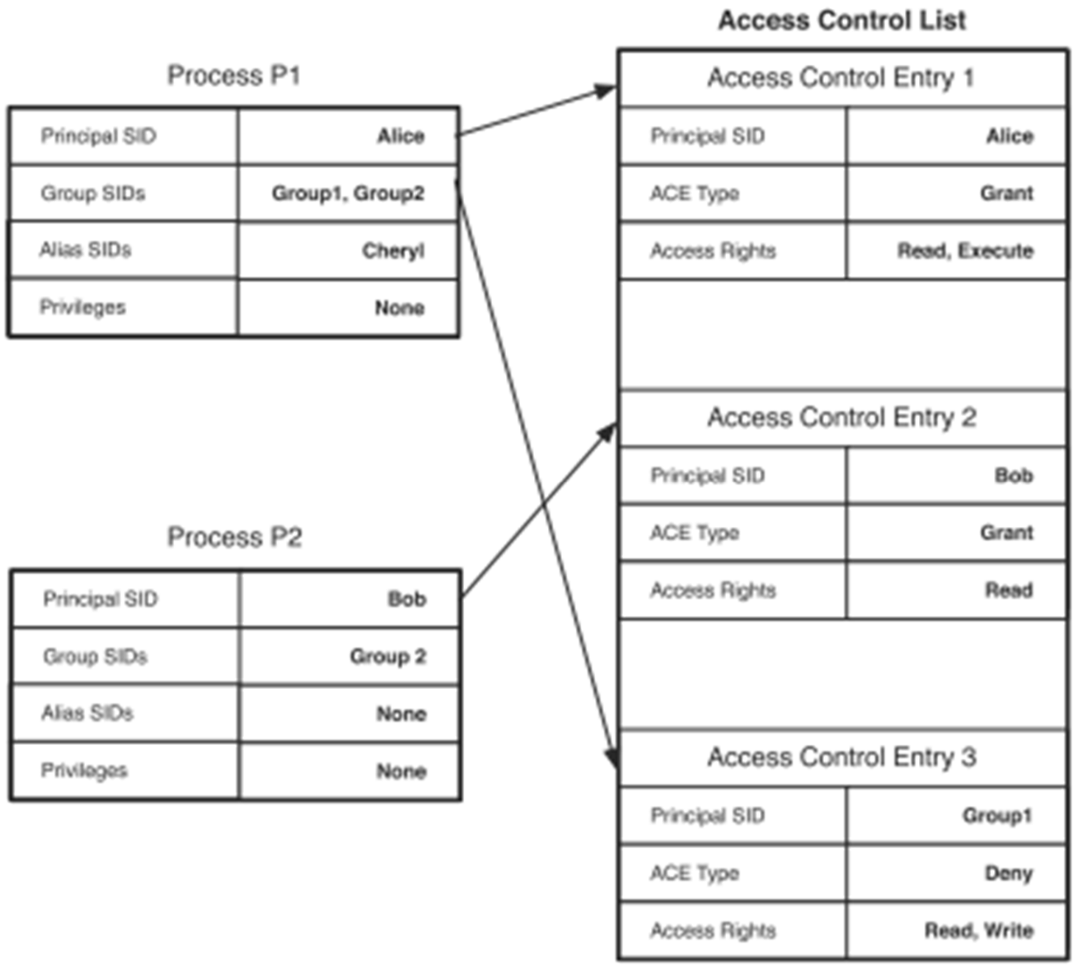
- Also using DAC model
- Subjects in Windows are similar to subjects in UNIX, each process is assigned a token that describes the process's identity.
- A process identity consists of user security identifier (principal SID,~ UNIX UID), a set of group SIDs (not single GID), a set of alias SIDs (on behalf of another identity), and a set of privileges
- A Windows identity is still associated with a single user identity, but a process token for that user may contain any combination of rights.

WINDOWS PROTECTION SYSTEM (2/3)

- Windows objects can belong to a number of different data types besides files.
- Applications may define new data types, and add them to the ***active directory***,
- Active durectory is the hierarchical name space for all objects known to the system
- Windows supports arbitrary access control lists (ACLs) rather than the limited mode bits approach of UNIX

WINDOWS PROTECTION SYSTEM (3/3)

- A Windows ACL stores a set of access control entries (ACEs) that describe which operations an SID (user, group, or alias) can perform on that object
- ACEs may either grant or deny an operation



WINDOWS AUTHORIZATION

- Windows authorization queries are processed by a specific component called the Security Reference Monitor (SRM)
- SRM is a kernel component
- The SRM uses the object SID to retrieve its ACL from which it determines the query result
- Above example:

P1, read: ok
P1, read, write: no
P2: read: ok
P2: read, write: no

WINDOWS VULNERABILITIES

- The Windows Registry
- Administrator Users
- Enabled By Default : example in IIS

OPERATING SYSTEMS HARDENING

Basic steps should be used to secure an operating system:

- Install and patch the operating system
- Harden and configure the operating 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 systems (IDS)
- Test the security of the basic operating system to ensure that the steps taken adequately address its security needs

Update System–Frequently

- Automatic Security Updates:
- Windows: turn on auto update or manual
- Linux:
  - CentOS uses ***yum-cron*** for automatic updates.
  - Debian and Ubuntu use ***unattended upgrades***.
  - Fedora uses ***dnf-automatic***.

Add a Limited User Account

- Accessed your Linux as the root user
- Creating a limited user account
- Administrative tasks will be done using sudo with the limited user account

1.Create the user, replacing example_user with your desired username, and assign a password

2. Add the user to the a group for sudo privileges:

Harden SSH Access: Create an Authentication Key-pair

- Using cryptographic key pair for more secure
- Create a key pair and configure to not accept passwords for SSH logins
- Windows:
  - This can be done using PuTTY as outlined in our guide: Use Public Key Authentication with SSH.
- Linux / OS X:
  - To check for existing keys, run `ls ~/.ssh/id_rsa*`
  - `ssh-keygen -b 4096`

- Upload the public key

-Linux:

```
ssh-copy-id username@ip adrress of server
```

-OS X

On your server

```
mkdir -p ~/.ssh && sudo chmod -R 700 ~/.ssh/
```

From local computer:

```
scp ~/.ssh/id_rsa.pub username@ip adress of server:~/.ssh/authorized_keys
```

-Windows: using WinSCP or PuTTY (Copy the public key directly from the PuTTY key generator into the terminal emulator connected to your server)

```
mkdir ~/.ssh; nano ~/.ssh/authorized_keys
```

- set permissions for the public key directory and the key file itself:

```
sudo chmod 700 -R ~/.ssh && chmod 600 ~/.ssh/authorized_keys
```

Linux: Disallow root logins over SSH

- /etc/ssh/sshd_config

```
# Authentication:
...
PermitRootLogin no
```

Linux: Disable SSH password authentication

- /etc/ssh/sshd_config

```
# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no
```

Linux:Listen on only one Internet protocol

- The SSH daemon listens for incoming connections over both IPv4 and IPv6 by default. Unless need to SSH into server using both protocols, disable whichever we do not need

AddressFamily inet to listen only on IPv4

Or
AddressFamily inet to listen only on IPv6

Add it to the end of the sshd_config file

Linux: Restart the SSH service to load the new configuration

```
sudo systemctl restart sshd
```

or

```
sudo service ssh restart
```

Linux: Remove Unused Network-Facing Services

- Determine Running Services

```
sudo netstat -tulpn
```

(install the package net-tools in oder to run netstat command)

- Determine Which Services to Remove

Linux: Uninstall the Listening Services

- How to remove the offending packages will differ depending on your distribution’s package manager.

```
sudo yum remove package_name

sudo apt-get purge package_name

sudo dnf remove package_name
```

Linux: Configure a Firewall

- View Your Current iptables Rules

```
sudo iptables  –L
sudo ip6tables  -L
```

- Basic iptables Rulesets for IPv4 and IPv6
- Verify iptables Rulesets

```
 sudo iptables  –vL
sudo ip6tables  -vL
```

Install IDS

- Host IDS
- Network IDS
- Example Tripwire, snort

Hardening Windows Server

- Refer documents such as Hardening Windows Server 2008, 2012….
- Refer

https://cyber-defense.sans.org/blog/2009/08/12/blue-team-defender-guide-cyber-war-games