CSI 4139 Notes

Operating System Security

Operation system security layers

- each layer of code needs measures in place to provide appropriate security services
- each layer is vulnerable to attack from below if the lower layers are not secured properly

Top 4 measures for prevention

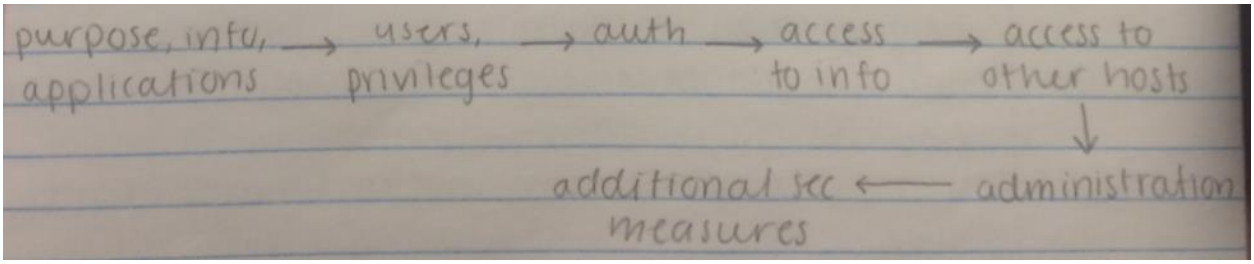
- Patch OSs and applications using auto-update
- Patch third-party apps
- white-list approved apps
- restrict admin privilege to users who need them

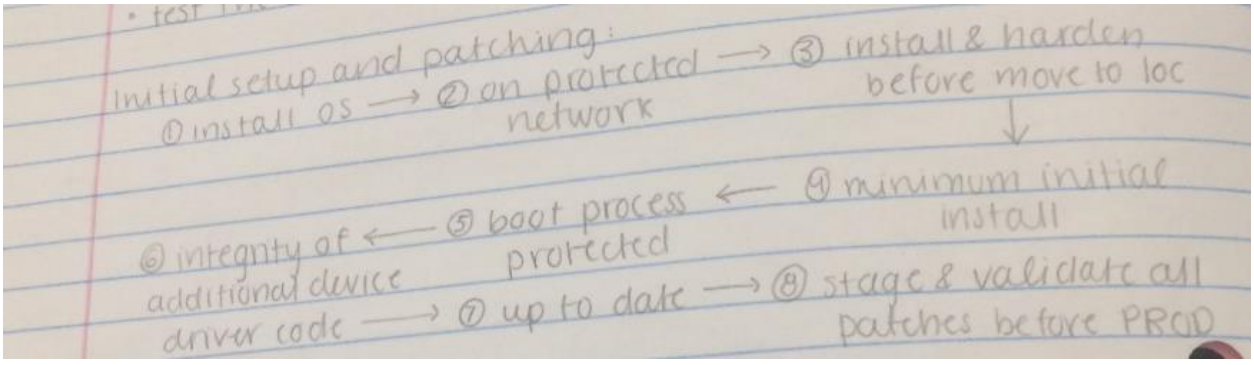
It's possible for a system to be compromised during the installation process before it can install the latest patches:

- secure the underlying OS then the key apps
- assess risks and plan the system deployment
- ensure any critical content is secured, appropriate network protection mechanisms are used and appropriate process are used to maintain security

Planning:

- Include a wide security assessment at the organization
- Aim to maximize security while minimizing costs
- Process needs to fins security requirements or the system, apps, data and more
- Identify appropriate personnel training to install and manage



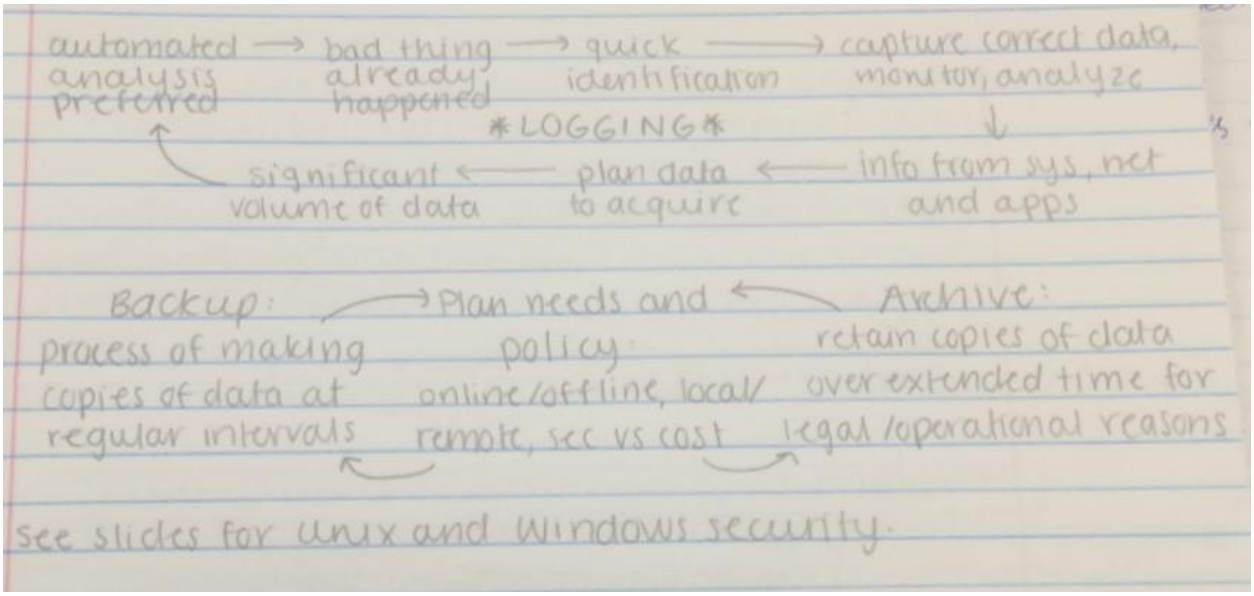


Remove unnecessary sources, applications & protocols

- fewer software packages = reduces risk
- planning process should identify, do not use defaults
- when performing initial install, do not use defaults → maximizes case of use rather than security
- secure or disable default accounts

Configure resources controls: permissions on data and res install additional sec control: anti-virus sw, firewalls, IDs or IPs allocation white-list.

Test the security: ensure good config and identify vulnerabilities after hardening → repeat periodically as part of sec maintenance.



Virtualization:

- Technology that provides an abstraction at the recourses used by some software which runs in a simulated env → VM
- Benefits include better efficiency in the use of the physical system resources
- Provides supports for multiple Oss on one physical system
- App virtualization → full virt → VM monitor (many VMs)

Virtualization security concerns:

- Guest OS isolation only access and use its resources
- Guest OS monitoring by hypervisor (VMM): privileged access
- Virtualized env sec: image and snapshot mgmt. under attack
- Secure all elements
- Hypervisor sec: install in isolated env, configure local and remote administration, secure remote with firewall and IDs, admin traffic on separate network with limited access

Malware

NIST program inserted into a system with the intention of damaging or disabling it or annoying the user → more for book 6.1

Malware

- Used to be for fun, now done by governments crime organizations and enterprises.
- It takes skill to develop malware but no skills to use it
- Hard to find the originator, harder to prosecute

Malicious Logic

- Trojan horse: does something good while also secretly doing something bad
- Worm self-replicates → virus replicates using executable specifically

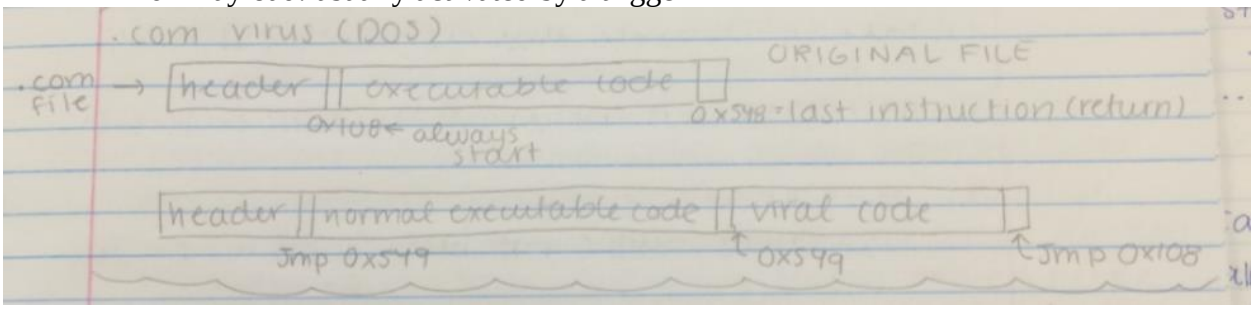
Malicious code has been around since the 1960s, when a university would have CPV and terminals.
It is extremely hard to guarantee that a system is free of malicious code → it could be in the compiler code.

1988: the internet worm hid itself with encryption used a weakness in the mall daemon.

- It replicated even if the system was infected, and tool down the internet for a few days

How do viruses/worms work?

- In two parts: Replication mechanism and payload
    - Replication: worm makes a copy of itself somewhere else when it runs
    - Payload: usually activated by a trigger



➔ Payload (ctd): make changes to data, steal resources, take down network, stealing or publishing user data, create a backdoor, perform benevolent software upgrades

Note: concern isn't always malicious code, malicious data is also a problem
➔ Example: takes a desmopressin server down by sending it many super compressed files

Types of viruses:

- Boots sector infector
- Encrypted virus usually all exception decryption routine and key
- Executable infector
- Terminate and stay resident stays in memory
- Stealth virus: primary goal is to stay hidden, which it does by sending take info to OS

Malware defenses/countermeasures

- First step up to date software proper AC
- Antivirus software
    - Host based scanners: 4 generations
        - Simple scanners: check for virus signature which virus uses not to re-infect and inoculates files
        - Network based scanners: at network periphery after on email server
            - Mgress: in
            - Egress: out
            - Both above are block suspicious packets → prevent botnet activity
        - Distributed intelligence gathering approaches:
            - Send all info to central system for analysis, updating signatures (Many clients)
        - Rootkit detection: checksummers (unauthorized changes to files), with values sotre in protected space and gotten from clean files.

Information Flaw Security

Information flow is the way information moves through a system. We want to preverve confidentiality or integrity of data
Conf: prevent flow to user not authorized to receive it
Integ: prevent flow to a process more trustworthy than a data

- Protecting info flow through programs
- A command sequence c causes a flow of info from x to y if, after the excuction of x, some info about x before x was executed can be deduced from the value of y after c was executed
- Explicit → via direct assignment
- Implicit: → otherwise
- If x is a variable, then x is the info flow class of x (security label, integrity label, category)
- Compiler-based mechanism check that info flows throughout a program are authorized:
- Information may flow from x to y if x <= y to preserve continentality, or if x >= y preserve integrity