Intro to Computer Security

Friday, January 17, 2025 5:43 PM

In terms of a secured computer
- We can see only what we should be able to see
- We can't modify what we should not be able to modify
- We can verify that what we are seeing is from the entity we expect
- We can access the things that should be available
- We are protected : Hackers, failures, negligence, misuse

Confidentiality, Integrity, Availability - CIA Triad
- Confidentiality
  - Only those authorized to view the data can actually view it
- Integrity
  - Data is not modified from an unauthorized source so we can ensure it
- Availability
  - Systems and services are available and resilient against attacks, failures and compromise

Applying Security to data involves
- Data at rest
  - Stored on filesystem, records in DB, physical folders
- Data in motion
  - Data communicated through the web, data sent via network, radio data via wiFi, bluetooth, etc

Threats
- Something that can cause harm to our software

Vulnerabilities
- Where a system or computer is vulnerable to harm

Risks
- The possibility of a threat happening
- Well known vulnerability in popular library increases risk

Managing Risk
- Identify Assets, Identify threats, Asses vulnerabilities, assess risks, and then mitigate them

Defense in Depth
- When implementing computer security, we want layers so this means we don't rely on a single defense

- Possible layers
  - External Network : VPN, DMZ, logging, pen testing, etc
  - Network Perimeter : Firewalls, proxies, logging, pen testing, etc
  - Internal Network - IDS, IPS, logging, pen testing
  - Host - Auth, hardening, IDS, IPS, …
  - Application - Patching, pen testing, auditing, …
  - Data - Encryption, backups, authorization, …

Key takeaways

CIA is our compass
Protect both data at rest and in transit
Identify all threats, vulnerabilities, risks
Some business may have to give way to risks, properly document them
Have a plan for the unexpected
Layers of defense

LINUX and CLI commands

Pwd - lists present working directory
Ls [path] - list files in a current directory or in a given path. If we pass -l, this gives us more info on the files. -a lists hidden files
Cat <file> - show contents of file
Set - will list variables set, -o will list options set
Cd - will change directory

.profile is a file that runs when we first login and we can put anything there

Understanding Paths
. - refers to current directory
.. - refers to one directory up
~ - refers to home directory of current user
/ - refers to root directory
* - Wildcard, can be mixed in pathnames or filenames

If a file has . Infront of it, it is a hidden file

If we had a file called test.c, we can list out the contents in the following ways

Cat test.c
Cat ../krupp/test.c
Cat ~/test.c
Cat /home/krupp/test.c

Other common commands

Grep <search> [filename] - used to search contents within file or from output
Less [filename] - used to scroll through file
Man command - pulls up manual for particular command

Piping Output to Input Process

Piping - take the output of one command and send it as the input to another command.

Ex: ps -ef | grep 212 <- Will look for a specific PID
Cat test.c | grep // | grep -v "Brian" <- Pulls up any comments in c source that do not have brian

in it

Redirecting Input/Output

Ps -ef > process_log - Takes the process listing and saves it in a file called process_log

S -ef >> process_log - takes the process listing an appends it to the file called process_log

Sending a Process to the background

Ps -ef &

We can redirect the output to a file as well : ps -ef > out &

If there is an error, we can also redirect that error

Ps -ef > out 2>&1 & 2-STDERR, is being appended to STDOUT(1)

To list processes in the background, type jobs

Looking at files

We can look at the bottom or top of a file using tail and head respectively

Tail -f logfile monitors new inputs on a file

File Permissions

To list file permissions, type ls -l

3 main permissions in Unix/Linux : read, write, execute (r,w,x).

```
-rw-r--r-- 1 root   operator 43   Sep    2 15:26 out
drwxr-xr-x 2 root   operator 128  Sep    2 15:16 somedir
-rwxr-xr-x 1 root   operator 5305 Sep    2 15:01 test
```

The first field is what type of file we are working with. The two most common types are - for normal file, and d for directory.

The next 3 letters are the permissions for the owner, so for file test, root has read, write, and execute permissions. Execute meaning they can run that file as code. Group has just read and execute permissions, so they cannot modify the file, and everyone else has the same permissions as the group for this case.

If you want to set permissions you use the chmod command, here are two ways of doing it. One is absolute where you specify the exact permissions you want set by specifying the number for each permission area, User, Group, Everyone else. Each number corresponds to the appropriate bit, rwx, and are computed as a binary number. Read = 4, Write = 2, Execute = 1. So if I want rwx for the owner and no permissions for everyone else, I would do the following on a file:

The command chmod 760 is for rwx for owner, read and write to group, and nothing for everyon else

7 - 4(read) + 2(write) + 1(execute)
6 - 4 (read) + 2 (write)
0 - nothng

We can also use chmod to change permissions relative to what a file currently has set

`chmod u+x test` Adds execute permission for the user
`chmod +x test` Adds execute permission for everyone
`chmod g+rw` Adds read/write for group
`chmod o-x` Removes execute for other (everyone that is not the owner or the group

Moving files around

We can move files around, rename them, and create directories with them

`mv <file> <dest>` Moves a file to a directory or renames it
`rm <file>` Removes a file
`mkdir <dirname>` Makes a directory
`rmdir <dirname>` Removes a directory
`rm -fr directory_name` Removes a directory and everything in it (be careful!)
`cp <source file> <destination>` Copies a file or directory

As an example using wildcards, if you wanted to remove all programs that end with .c in a file, you would do the following: `rm *.c`
exit

Important Environment variables
PATH - Where we look for a particular command
PS1 - What you see in the shell

To set an environment variable
PS1 = "readytorun:"

To print out environment variable
Echo $PS1

Other Useful commands
Uname - list information on OS you are on,
P addr - List information about the network interface and P
Host <name> - Retrieves IP for a given name
Hostname <name> - Sets the hostname or get the hostname of your OS
Cat /etc/os-release - Shows information about the current OS
Locate <name> - Pass in the name of a file to help locate ti.