Linux Security and Hardening Essential Training

General Security

- Linux is "secure", but it's not a panacea.
- People play a key role in security.
- Security is an ongoing process.
- Linux security features
  - Open Source.
  - It's not a popular target.
  - Package management.
  - Separation of privileges(multi-user system).
- Security Principles
  - Principles of Least Privilege
  - Use encryption
  - Shared accounts (Yes, root can be a shared account!)
  - Multi-factor authentication
  - Firewall
  - Monitoring logs

Physical Security

Protect from Grub `e` edit single user mode

- In systemd go to `/lib/systemd/system/`
- Replace `sushell` with `sulogin` in emergence.service and rescue.service

Protect Grub by password

- Let's just say username=`grubProtect` and password=`grubzilla123`
- In `/etc/grub.d/40_custom` add

```
set superuser="grubProtect"
password grubProtect grubzilla123
```

- For encrypted password

```
grub-mkpasswd-pbkdf2
```

- Enter the password=`grubzilla123` and get an output like this

```
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.1CAEF371E5B24AF502560031A265F2
```

- In /etc/grub.d/40_custom

```
set superuser="grubProtect"
password_pbkdf2 grubProtect grub.pbkdf2.sha512.10000.1CAEF371E5B24AF502560031A265F29E05
```

- Then `update-grub`

Disk Encryption

- Install `cryptsetup`
- Encrypt New Device/Disk

  Encrypting Disk Will Remove All Data

  - Fill device with random data

    ```
    sudo shred -v -n 1 <e.g. /dev/sdb , i.e diskname>
    ```

  - Now Run `cryptsetup` and put in the passphrase

    ```
    sudo cryptsetup luksFormat <e.g. /dev/sdb , i.e diskname>
    ```

  - Open the device and put in the passphrase

    ```
    sudo cryptsetup luksOpen <e.g. /dev/sdb , i.e diskname> <e.g. opt, i.e name for the
    ```

  - Since here we named the folder `opt`, you can find that disk on `/dev/mapper/opt`

  - Formating the device

    ```
    sudo mkfs -t ext4 /dev/mapper/opt
    ```

  - Close the device

    ```
    sudo cryptsetup luksClose opt
    ```

  - Updating `/etc/fstab` for mounting while login

    ```
    /dev/mapper/opt /opt ext4 defaults 0 0
    ```

  - Updating `/etc/crypttab` for asking passphrase while mounting

    ```
    opt /dev/sdb none luks
    ```

- Encrypt a File or Folder

  - For example Make folder `/data`

    ```
    sudo mkdir /data
    ```

  - Locate 100mb to a file `opt` in `/data`

    ```
    sudo fallocate -l 100M /data/opt
    ```

  - Adding random data to file `opt` `sudo dd if=/dev/urandom of=/data/opt bs=1M count=100`

  - To check the random data in `/data/opt`

    ```
    sudo strings /data/opt
    ```

  - Now encrypting `/data/opt`

    ```
    sudo cryptsetup luksFormat /data/opt
    ```

  - Open

    ```
    sudo cryptsetup luksOpen /data/opt opt
    ```

  - Format

    ```
    sudo mkfs -t ext4 /dev/mapper/opt
    ```

  - Mount

    ```
    sudo mount /dev/mapper/opt /opt
    ```

- Encrypting Device with Data
  - Backup this Device
  - Fill the Device with random data using `shred` or `dd`
  - Encrypting the Device
  - Open the Device
  - Format it, mostly `ext4`
  - Mount and use it

Disable `Control` + `Alt` + `Delete`

- `Control` + `Alt` + `Delete` in systemd, rebooting your system.
- To disable this

  ```
  systemctl mask ctrl-alt-del.target
  systemctl daemon-reload
  ```

Summary

- Physical security threats.
- Physical security guidelines.
- Single user mode defenses.
- Kernel Parameter Protection.
- Disk encryption with LUKS.
- Disabling reboots from `Ctrl+Alt+Del`

Account Security

PAM (Pluggable Authentication Modules)

- Location: `/etc/pam.d` `/etc/pam.d/login` `/etc/pam.d/sshd`
- Format:

  ```
  module_interface
  control_flag
  module_name
  module_args
  ```

- PAM Modules Interfaces

  - `auth` - Authenticates users.
  - `account` - Verifies if access is permitted.
  - `password` - Changes user's password.
  - `session` - Manages user's sessions.

- PAM Control Flags
  - `required` - Module result must be successful to continue.
  - `requisite` - Like required, but no other modules are invoked.
  - `sufficient` - Authenticates user if no required modules have failed, otherwise iggnored.
  - `optional` - Only used when no other modules reference the interface.
  - `include` - Includes configuration from another file.
  - complext control flags - `attribute=value` for more info `man pam.d`

- PAM configuration
  - Configuration:

    ```
    account required pam_nologin.so
    session required pam_unix.so
    ```

  - Getting Help:

    ```
    man pam_nologin
    man pam_unix
    ```

Linux Account Types

- `root`, the superuser
  - Root can do anything.
  - Always has the UID of 0.
- System Accounts
  - UIDs of System Accounts are < 1000
  - Configured in `/etc/login.defs`
  - `useradd` with `-r` flag specifies to have UID of System Account Range
    `useradd -r system_account_name`
- Normal User Accounts
  - UIDs of Normal User Accounts are 1000
  - Intended for human (interactive) use
- Password Security
  - Enforcing, not hope for, strong passwords.
  - Protect against weak use pam_pwquality, based on pam_cracklib.
    * Configuration File:
      `/etc/security/pwquality.conf`
    * PAM Usage:
      `password requisite pam_pwquality`
    * Module attributes:
      `man pam_pwquality`

  - Use Shadow Password
    * Usually encrypted passwords is stored in `/etc/passwd`
    * But `/etc/passwd` is accessible with all users
    * When shadow password is enabled, the passwords at `/etc/passwd` is replaced with `x`
    * And the passwords are stored in /etc/shadow which is only accessible by root or the superuser
    * Converting regular passwords to shadow passwords
      `pwconv`
    * Converting shadow passwords to regular passwords
      `pwunconv`
    * Each feild of `/etc/shadow` is separated by `:`
      · Username
      · HashedPassword
      · DaysSinceEpochOfLastPasswordChange
      · Days until change allowed
      · Days before change required
      · Days warning for expiration
      · Days before account inactive
      · Days since epoch when account expires
      · Reserved
  - User account expire info
    ```
    chage -l <account>
    ```

Controlling Account Access

- Locking and Unlocking Accounts
  - Locking

    ```
    passwd -l <account>
    ```

  - Unlocking

    ```
    passwd -u <account>
    ```

- Locking using `nologin` as Shell

  ```
  chsh -s <shell> <account>
  chsh -s /sbin/nologin <account>
  ```

Account Security

- Disable `root` Logins
  - Update `/etc/pam_securetty`

    ```
    auth [user_unknown=ignore success=ok \
    ```

```
ignore=ignore default=bad] pam_securetty.so
```

  * `/etc/securetty` contains list of devies where `root` is allowed to login
    · Example

```
console
tty1
```

- Disable SSH root Logins
  * Update `/etc/ssh/sshd_config`

```
PermitRootLogin no
```

  * Then reload `systemctl`

```
systemctl reload sshd
```

- System/Application Accounts
  - Use one Account per service

    webs service (httpd), web service account (apache)
  - Don't allow direct logins from the account
    * Update `/etc/ssh/sshd_config`

```
DenyUsers <account1> <account2> ... <accountN>
```

  - Use `sudo` for all access

```
sudo -u apache apachectl configtest
```

- Delete Accounts
  - Determine the UID

```
id <account>
```

  - Delete Account

```
userdel -r <account>
```

    * The `-r` flag removes the home directory with removing user
  - Find files belonging to that account

```
find / -user <UID>
findn / -nouser
```

Network Security

Securing Network Services

- Use a dedicated user for each service.

  Take advantage of privilege separation.

- Ports below 1024 are privileged.

  Use root to open them, then drop privileges.

  Configuration Controlled by each service.

- Stop and unistall unused services
- Avoid unsecure services

  Use SSH instead of telnet, rlogin, rsh, and FTP

- Avoid information leakage or revealing information where possible
  - Web server banners.
  - Files `/etc/motd`,`/etc/issue`, and `/etc/issue.net`
- Disable and Uninstall services that are not required.
- List Listening Programs with netstat

  ```
  sudo netstat -nutlp
  ```

- Port Scanning using `nmap`

  ```
  nmap <hostname/IP>
  nmap localhost
  nmap 10.11.123.23
  ```

- List open file using `lsof`

  `-i` flag shows all listening network files

  ```
  lsof -i
  ```

- Testing a Specific Port
  - Using `telnet`

    ```
    telnet <hostname/IP> <port>
    ```

  - Using `nc`

    `-v` flag means its running in verbose mode

    ```
    nc -v <hostname/IP> <port>
    ```

- Xinetd Controlled Services
  - Services Controlled by Xinetd could be found in `/etc/xinetd.d/` folder
  - To disable a service update service config file

    ```
    disable=yes
    ```

SSH (Secure SHell)

- Allows for key based authentication.
- To allow key-authentication update `/etc/ssh/sshd_config`

  ```
  PubkeyAuthentication yes
  ```

- Create SSH Keys
  - Use the `ssh-keygen` command to create a key.
  - You can create a passphrase for the key.
  - The `ssh-keygen` command creates `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.
- Adding the Public Key to Remote Host
  - To copy the key, use `ssh-copy-id`:

    ```
    ssh-copy-id <user>@<host>
    ```

  - This adds public key to `~/.ssh/authorized_keys`
- Force only Key Authentication
  - Updating `/etc/ssh/sshd_config`

    ```
    PasswordAuthentication no
    ```

  - This allowing only authentication with keys
  - Hence, protecting `ssh` from brute force attacks
- Controlling Root Logins
  - Updating `/etc/ssh/sshd_config`
  - To disable root logins

    ```
    PermitRootLogin no
    ```

  - To only allow root to login with key

    ```
    PermitRootLogin without-password
    ```

- Allow Only Certain Users and Groups SSH Access
  - Updating `/etc/ssh/sshd_config`

    ```
    AllowUsers <user1> <user2> ... <userN>
    AllowGroups <group1> <group2> ... <groupN>
    ```

- Deny Certain Users and Groups SSH Access
  - Updating `/etc/ssh/sshd_config`

    ```
    DenyUsers <user1> <user2> ... <userN>
    DenyGroups  <group1> <group2> ... <groupN>
    ```

- Comman `ssh` flag

- `-L` for SSH Port Forwording

  Basically using host machines port as the host machines by the client

  ```
  ssh -L <client port>:<host IP>:<host port> <user@host/dns>
  ssh -L 3306:127.0.0.1:3306 server1
  ssh -L 8080:www.google.com:80 server1
  ```

- `-D` for Dynamic Port Forwarding / SOCKS

  Basically forwarding all request to the client port to host port

  ```
  ssh -D <client port> <user@host/dns>
  ssh -D 8080 server1
  ```

- `-R` for Reverse Port Forwarding

  Basically forward all request from host machine back to client machine

  ```
  ssh -R <host port>:<host IP>:<client port> <user@host/dns>
  ssh -R 2222:127.0.0.1:22 server1
  ```

- Disable TCP Port Forwording
  - Updating `/etc/ssh/sshd_config`

    ```
    AllowTcpForwarding no
    GatewayPorts no
    ```

- Use SSHv2 instead of SSHv1
  - Updating `/etc/ssh/sshd_config`

    ```
    Protocol 2
    ```

- Bind SSH to a Specific Address
  - Updating `/etc/ssh/sshd_config`

    ```
    ListenAddress <host/address1>
    ListenAddress <host/address2>
    .
    .
    .
    ListenAddress <host/addressN>
    ```

- Change the Default SSH Port
  - Update `/etc/ssh/sshd_config`

    ```
    Port 2222
    ```

- Adding New Port to SELinux

  ```
  semanage port -a <SSH port> -p tcp <new port>
  semanage port -l | grep ssh
  ```

- Avoid Information Leakage
  - Disable the Banner

    Banner data that is usually stored at `/etc/issue.net` which is sent a remote user before authentication is allowed

    * Update `/etc/ssh/sshd_config`

      ```
      Banner none
      ```

- To Reload the Configuration

  ```
  systemctl reload sshd
  ```

- More Info

  ```
  man ssh
  man sshd
  man sshd_config
  ```

Linux Firewall

- Firewalls control network access.
- Linux firewall = Netfilter + IPTables
- Netfilter is kernal framework.
- IPTables is packet selection system.
- Use the `iptables` command to control the firewall.
- Default Tables
  - Filter
    * Most commanly used table.
    * It is used to block incoming or deny outgoing connections
  - NAT
    * Network Address Translation.
    * It allows a single IP address to be shared
  - Managle
    * Alter packets.
  - Raw
    * Rarely used
    * Used to disable connection tracking
  - Security
    * Used for manadatory access control networking rules
    * Used by SELinux
- Default Chains
  - INPUT
  - OUTPUT

  - FORWARD
  - PREROUTING
  - POSTROUTING
- Tables vs Chains

	INPUT	OUTPUT	FORWARD	PREROUTING	POSTROUTING
Filter	x	x	x		
NAT	x	x		x	x
Mangle	x	x	x	x	x
Raw		x		x	
Security	x	x	x		

- Rules
  - Rules = Match + Target
  - Match on
    - Protocol
    - Source/Destination IP or Network
    - Source/Destination Port
    - Network Interface
    - Eg: `protocol:TCP, source IP:1.2.3.4, destination port:80`
  - Targets
    - Chain
    - Built-in targets
      - ACCEPT
      - DROP
      - REJECT
      - LOG
      - RETURN

Command-Line interface

- Command `iptables/ip6tables`
  - `iptables` for IPv4.
  - `ip6tables` for IPv6.
  - List/View
    - Display the filter table.

    `iptables -L`

    - Display the NAT table.

`iptables -t nat -L`

  * Display using numeric output.

`iptables -nL`

  * Display using verbose output.

`iptables -vL`

  * Use line numbers.

`iptables --line-numbers -L`

- Creating and Deleting a Chain
  * Create Chain

    `iptables -t <table> -N <chain>`

  * Delete Chain

    `iptables -t <table> -X <chain>`

- Appending, Inserting, and Deleting Rules
  * Appending Rule
    · For appending a rule in the end of chain.

    `iptables -A <chain> <rule-specification>`

    · To specify table use `-t` flag, if not default is filter table.

    `iptables -t <table> -A <chain> <rule-specification>`
  * Inserting Rule
    · For inserting rule at the beginning of the chain.

    `iptables -I <chain> <rule-specification>`

    · After specifing the chain add a rule number to specify where the rule need to be inserted, if not it will default at 0th position

    `iptables -I <chain> <rule-number> <rule-specification>`
  * Deleting Rule
    · For deleting a rule from the chain

    · A rule can be deleted by specify the rule itself or rule number

    `iptables -D <chain> <rule-specification>`

    `iptables -D <chain> <rule-number>`
  * Flushing Rules
    · To delete all rules for a chain

    `iptables -t <table> -F <chain>`

- Rule Specification Options

Option	**Description**
-s Source IP -s 10.11.12.13 -s 10.11.12.0/24	Source IP, Network or Name.*Name is resolved when the rule is added.*

Option	Description
-d Destination IP -d 10.11.12.13 -d 10.11.12.0/24	Destination IP, Network, or Name
-p Protocol -p tcp -p udp -p icmp	Protocol
-m Module module_options	Enable extended packet matching module.(man iptables-extensions)
-p Protocol -m Protocol –sport Port -p tcp -m tcp –sport 8080 -p tcp –sport 8080	Source Port
-p Protocol -m Protocol –dport Port -p tcp -m tcp –dport 80 -p tcp –dport 80 -p udp –dport 53	Destination Port
-p icmp -m icmp –icmp-type Type -p icmp -m icmp –icmp-type echo-reply -p icmp –icmp-type echo-reply -p icmp –icmp-type echo-request	ICMP packet type(iptables -p icmp -h)
-m limit –limit rate [/second/minute/hour/day]-m limit –limit-burst -m limit –limit 5/m –limit-burst 10-m limit ! –limit 5/s	Match until a limit is reached.–limit default is 3/hours–limit-burst default is 5/s = second/m = minute/h = hour/d = day! invert the match

  - Target/Jump
    - To specify a jump point or target

```
-j <target/chain>
-j ACCEPT   #Built-in target
-j DROP     #Built-in target
-j LOGNDROP #Custom chain
```

- To Save the Rules
  - In Debian and Ubuntu install `iptables-persistent`

```
apt install iptables-persistent
```

  - To save the rules

```
netfilter-persistent save
```

  - Rules and Configuration will be saved in `/etc/iptables`
- `iptables` Examples
  - Drop all connection from source IP of 10.0.0.124

```
iptables -A INPUT -s 10.0.0.124 -j DROP
```

  - `-A INPUT` means the rule is being added to the INPUT chain

- Accepts all TCP connection from source IP of 10.0.0.0/24 and destination port 22

  ```
  iptables -A INPUT -s 10.0.0.0/24 -p tcp -dport 22 -j ACCEPT
  ```

- Drops all TCP connection for destination port 22

  ```
  iptables -A INPUT -p tcp --dport 22 -j DROP
  ```

- To Protect against DOS attacks

  ```
  iptables -I INPUT -p tcp --dport 80 \
  -m limit --limit 50/min --limit-burst 200 \
  -j REJECT

  iptables -I INPUT -p tcp --dport 80 \
  -m limit --limit 50/min --limit-burst 200 \
  -m state --state NEW -j REJECT
  ```

TCP Wrappers

- Host-based network ACL system.
- Controls access to "wrapped" services.
- A wrapped service is compiled with libwrap support.
- To Print required shared libraries run `ldd <path to the binary of the file>`
- Can control access by IP address/Networks.
- Can control access by hostname.
- Transparent to the client and service.
- Used with xinetd.
- Centralized management for multiple network services.
- Runtime configuration.
- TCP Wrapper Configuration
  - Configuration files to TCP Wrappers are `/etc/hosts.allow` and `/etc/hosts.deny`
  - When TCP connection request received first `/etc/hosts.allow` is checked.
  - If match is found, access is granted.
  - If not then next `/etc/hosts,deny` is checked.
  - If match is found access is denied and log message will be written.
  - If there are no matches, access is granted.
- TCP Wrappers Examples

  - The format given below is valid for both `/etc/hosts.allow` and `/etc/hosts.deny`

```
# SERVICE(S) : CLIENT(S) [ : ACTION(S) ]
sshd : 10.11.12.13
imapd : www.example.com
sshd, imapd : 10.12.11.13
ALL : 10.9.8.12, .example.com, .admin.example.com
sshd : jumbox*.example.com, jumbox0?.example.com # Regex Matching
sshd : 10.11.12.
sshd : 10.
sshd : 10.11.0.0/255.255.0.0
sshd : /etc/hosts.sshd # Path to a file with list of host
imapd : ALL
```

  - In `/etc/hosts.allow`

```
# SERVICE(S) : CLIENT(S) [ : ACTION(S) ]
sshd : ALL EXPECT .hacker.net
```

- TCP Wrappers Logging

```
# SERVICE(S) : CLIENT(S) [ : ACTION(S) ]
sshd : 10.11.12.13 : severity emerg
sshd : 10.11.12.13 : severity local0.alert
```

  - In `/etc/hosts.deny`

```
# SERVICE(S) : CLIENT(S) [ : ACTION(S) ]
sshd : .hacker.net : spawn /usr/bin/wall "Attack in progress."
sshd : .hacker.net : spawn /usr/bin/wall "Attack from %a."
```

- Expansions

Format Specifiers	**Description**
%a (%A)	The client (server) host address
%c	Client information.
%d	The daemon process name.
%h (%H)	The client (server) host name or address.
%n (%N)	The client (server) host name.
%p	The daemon process id.
%s	Server information.
%u	The client user name (or “unknown”).
%%	Expands to single `%` charater.

File System Security

File and Directory Permissions

```
ls -l
-rwxrw-r-- user:group bytes data time filename
```

- Permission - Files vs Directories

Symbol	Permission	File	Directory
r	Read	Allows a file to be read.	Allows file names in the directory to be read.
w	Write	Allows a file to be modifed.	Allows entries to be modified within the directory.
x	Execute	Allows the execution of a file.	Allows access to contents and metadata for entries.

- Permission Categories

Symbol	Categories
u	User
g	Group
o	Other
a	All

- Groups
  - Every user is at one group.
  - Users can belong to many groups.
  - Groups are used to organize users.
  - The `group` command displays a user's groups.
  - You can also use `id -Gn`
- Secret Decoder Ring

Type	User	Group	Other
1	3	3	3
d-	rwx	rwx	rwx

  - In Type
    * Directory is denoted by `d`.
    * File is denoted by `-` or `.`
  - In Users, Groups, and Others the order of permission is (Read , Write m Execute) rwx
  - If any permission is not granted to a file or folder the charater of the permission gets replaced with `-`

- Changing Permission

Item	Meaning
chmod	Change mode command
ugoa	User categoryuser ,group, other, all
+-=	Add, subtract, or set permissions
rwx	Read, Write, Execute

```
chmod u+w filename # User is granted with Write permission
chmod u-rw filname # User is denied of Read and Write permission
chmod u+rwx,g-x,o-rwx filname # User is granter with rwx, Groups are denied of x, and O
chmod a=r filname # All (User, Groups, and Others) are Granted with only r permission
chmod u=rwx,g=rx,o= filename # User has rwx, Groups have rx and Others has None
```

- Numeric Based Permissions

	r	w	x
Value for off	0	0	0
Binary value for on	1	1	1
Base 10 value for on	4	2	1

  - Possible Numeric Permissions

Octal	Binary	String	Description
0	000	---	No permissions
1	001	--x	Execute only
2	010	-w-	Write only
3	011	-wx	Write and Execute (2+1)
4	100	r--	Read only
5	101	r-x	Read and Execute (4+1)
6	110	rw-	Read and Write (4+2)
7	111	rwx	Read, Write, and Execute (4+2+1)

- Working with Group

  - New files belong to your primary group.
  - The `chgrp` command changes the group.

  `chgrp group filename/directoryname`

- Directory Permissions Revisited
  - Permissions on a directory can affect the files in the directory.
  - If the file permissions look correct, start checking directory permissions.
  - Work your way up to the root
- File Creation Mask
  - File creation mask determines default permissions.
  - If no mask were used permissions would be:
    * 777 for directory
    * 666 for files
- The `umask` Command

  `umask [-S] <mode>`

  - Sets the file creation mask to mode, if given.
  - Use -S t symbolic notation.

    ```
    |   |Directory|File|
    |---|:---:|:---:|
    |Base Permission|777|666|
    |Subtract Umask|\-022|\-022|
    |Creations Permission|755|644|
    ```

- Special Modes
  - umask 0022 is the same as umask 022
  - chmod 0644 is the same as chmod 644
  - The special modes are:
    * setuid
    * setgid
    * sticky
- Summary
  - Symbolic permissions.
  - Numeric/octal permissions.
  - File vs Directory permissions
  - Changing permissions.
  - Working with groups.
  - File creation mask.

Special Modes

File Attributes

ACLs

Rookit Hunter