

Day 36

Exploitation Analyst

User Management and PAM:

Disable root Login:

Why Root login is so crucial?

Root login in Linux is important because it grants unrestricted access to the entire system, including all files, configurations, and administrative commands. It allows full control for system management, software installation, user management, and troubleshooting. However, misuse can cause security risks, system instability, or data loss.

Which file contains the data for defining the root users?

In Linux, the file `/etc/sudoers` controls who can execute commands as the root user. It defines permissions for specific users or groups to run administrative commands with `sudo`, effectively allowing them root privileges without direct root login.

Which file contains the data to disable any root login?

The file `/etc/ssh/sshd_config` contains the setting `PermitRootLogin` that can be used to disable root login over SSH. Setting `PermitRootLogin no` in this file prevents remote root logins. root login can also be disabled by editing `/etc/passwd`. If you change the root user's shell from something like `/bin/bash` to `/sbin/nologin` or `/bin/false`, it prevents interactive logins for root, both locally and remotely, while keeping the account itself present for system processes.

Which one is better?

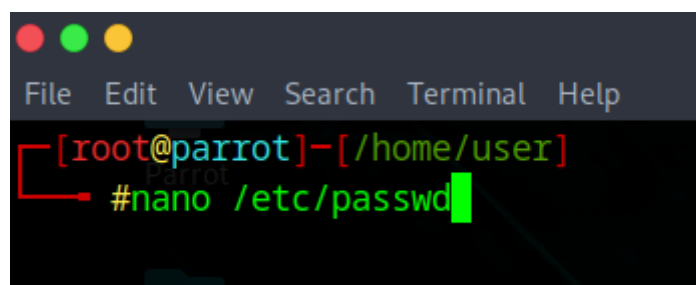
Disabling root login via `/etc/ssh/sshd_config` is generally better because it only blocks remote SSH access for root while keeping the account functional for local administrative use and automated processes.

Editing `/etc/passwd` is riskier, as it blocks all interactive logins for root and could disrupt recovery or maintenance tasks that require direct root access.

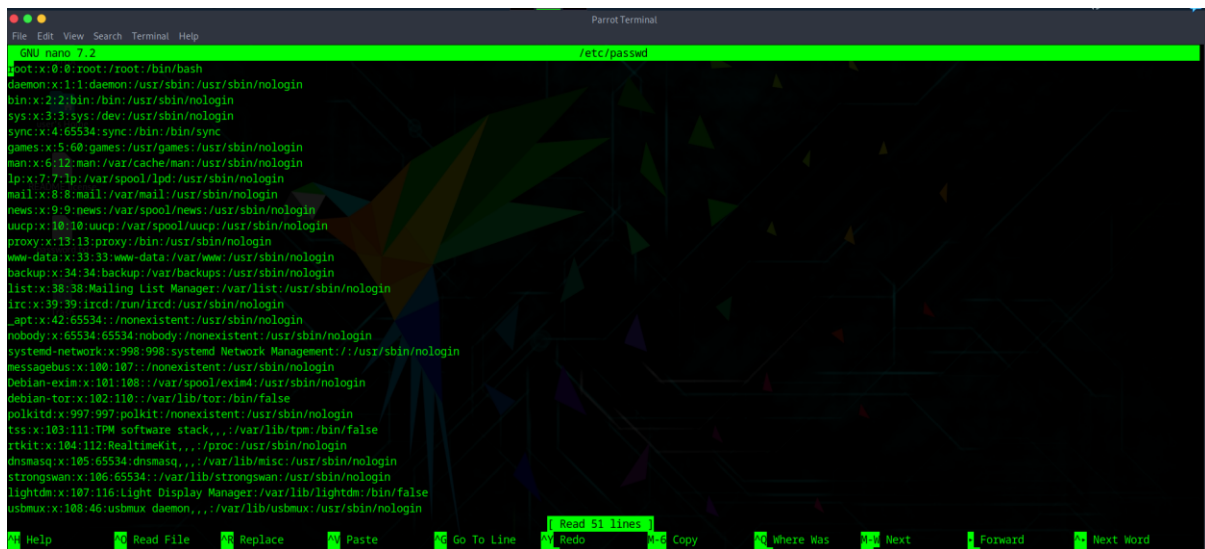
Disabling root Login:

Steps: (by `/etc/passwd`)

Open the file `/etc/passwd` using the nano command:

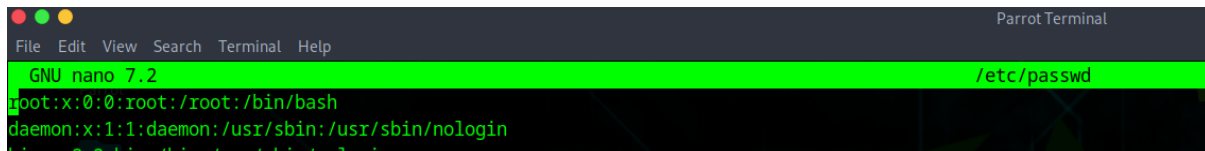
A screenshot of a terminal window with a dark background. At the top, there are three colored window control buttons (red, green, yellow) and a menu bar with the options 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. Below the menu bar, the terminal prompt shows the user is root on a machine named 'parrot', with the current directory being '/home/user'. The command '#nano /etc/passwd' has been entered, and a green cursor is positioned at the end of the command line.

Following screen will appear:



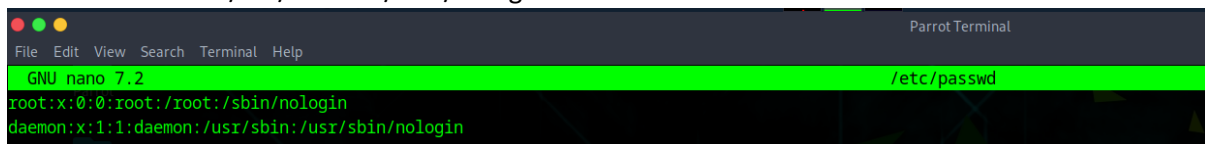
```
GNU nano 7.2 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
messagebus:x:180:180::/nonexistent:/usr/sbin/nologin
binfmt-x86:x:101:108:/:/var/spool/evms4:/usr/sbin/nologin
debiana-tor:x:102:110:/:/var/lib/tor:/bin/false
polkitd:x:997:997:polkit:/nonexistent:/usr/sbin/nologin
tss:x:103:111:TPM software stack,,:/var/lib/tpm:/bin/false
rtkit:x:104:112:RealtimeKit,,:/proc:/usr/sbin/nologin
dnsmasq:x:105:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
strongswan:x:106:65534:/:/var/lib/strongswan:/usr/sbin/nologin
lightdm:x:107:116:Light Display Manager:/var/lib/lightdm:/bin/false
usbmux:x:108:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
```

Focus on the part mentioning 'root':



```
GNU nano 7.2 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Edit that line from /bin/bash to /sbin/nologin:



```
GNU nano 7.2 /etc/passwd
root:x:0:0:root:/root:/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

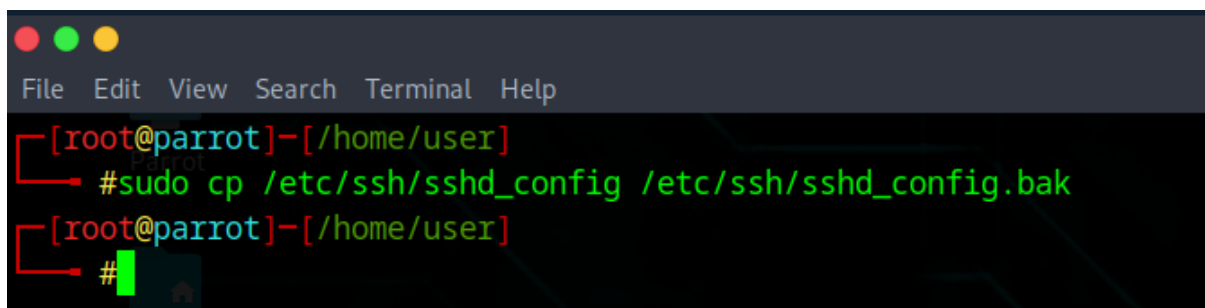
Save the file and exit.

Disabling root Login:

Steps: (by /etc/ssh/sshd_config)

Ensure you have a sudo-enabled user so you don't lock yourself out.

Backup the SSH config file: `sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak`

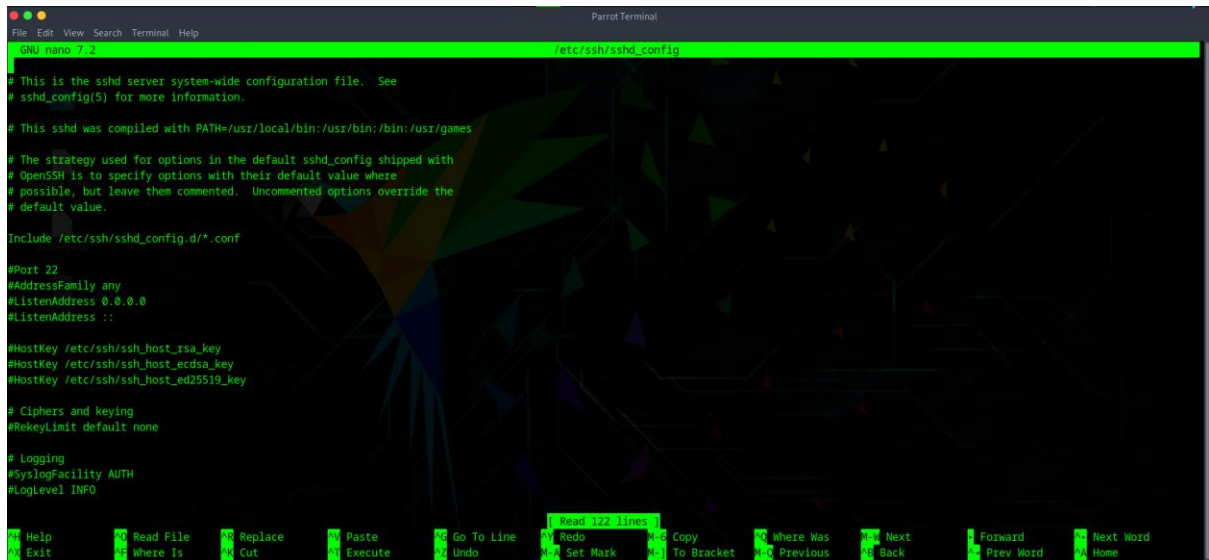


```
[root@parrot]-[/home/user]
#sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
[root@parrot]-[/home/user]
#
```

Open the file for editing: `sudo nano /etc/ssh/sshd_config`

```
#sudo cp /etc/ssh/sshd_config /etc/ssh/
[roor@parrot]-[/home/user]
#sudo nano /etc/ssh/sshd_config
```

Following screen will appear:



```
GNU nano 7.2 /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.
Include /etc/ssh/sshd_config.d/*.conf
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key
# Ciphers and keying
#RekeyLimit default none
# Logging
#SyslogFacility AUTH
#LogLevel INFO
```

Search for this section: PermitRootLogin

```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Change that section to: PermitRootLogin no

```
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Save it and exit. Reload SSH service: `sudo systemctl reload sshd`

Test SSH as root to confirm it's blocked.

How does editing these two files differs?

1. Editing /etc/ssh/sshd_config (PermitRootLogin)

- This only affects SSH remote logins.
- When you set PermitRootLogin no and reload sshd, the SSH daemon refuses any connection attempts using the root username.
- The root account still exists and can log in locally (e.g., via terminal, recovery mode, or su).
- Safer because you can still use root if you have local machine access.

2. Editing /etc/passwd (changing root's shell)

- /etc/passwd contains account information, including the shell to start after login.
- If you change root's shell to /sbin/nologin or /bin/false, it prevents all interactive logins for root, both locally and remotely.
- The root account still exists for running system processes, but you can't open a root shell directly.
- Riskier because if you lose your sudo access or other admin user, you could lock yourself out completely.

What we can do make sure that these settings works fine?

1. **Principle of Least Privilege** – Never allow direct root login over SSH unless absolutely necessary. Use a normal user with sudo for admin tasks.
2. **Fail-Safe Access** – Always have at least one other sudo-enabled account before disabling root login, so you don't lock yourself out.
3. **Layered Security** – Disabling root SSH login is one layer; combine it with strong passwords, SSH key authentication, and firewall rules.
4. **Method Selection** – Prefer /etc/ssh/sshd_config over /etc/passwd for root login restriction because it's less disruptive and easier to reverse remotely.
5. **Testing** – Always test changes in a separate SSH session before closing your current one, so you can fix issues if something goes wrong.
6. **Logging** – Monitor /var/log/auth.log (Debian/Ubuntu) or /var/log/secure (RHEL/CentOS) for any failed root login attempts after changes.
7. **Recovery Knowledge** – Know how to re-enable root login via local console or recovery mode in case of emergencies.

Note:

The biggest difference is scope:

- /etc/ssh/sshd_config → blocks only remote SSH root login.
- /etc/passwd → blocks all interactive root logins, both local and remote.

--The End--