Linux Server Hardening (Basic)

Description:

This report documents the basic hardening process applied to an Ubuntu 24.04 server running in a virtualized environment.

The goal of this project was to secure SSH access, enable firewall protection, configure automatic updates, and mitigate brute-force attacks using Fail2ban.

The following steps were implemented and successfully tested.

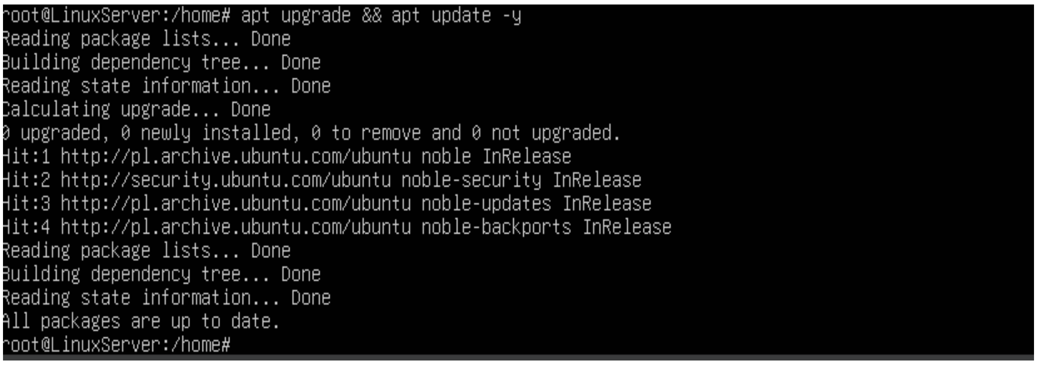
Environment

- **Operating System:** Ubuntu Server 24.04.3 LTS
- **Virtualization:** Oracle VirtualBox 7.1.12
- **Network Mode:** Bridged Adapter (LAN accessible)
- **Role:** Basic secured server for demonstration of hardening practices

Hardening Steps

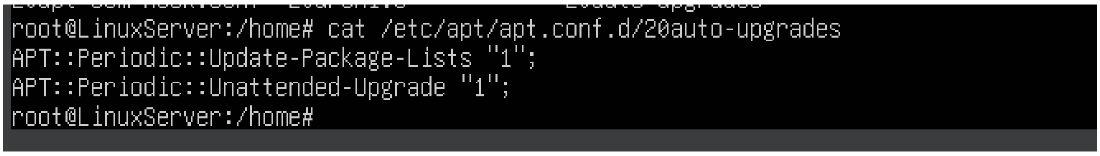
System Updates

Applied latest security patches with ` apt update && apt upgrade -y`



```
root@LinuxServer:/home# apt upgrade && apt update -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Hit:1 http://pl.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://pl.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://pl.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@LinuxServer:/home#
```

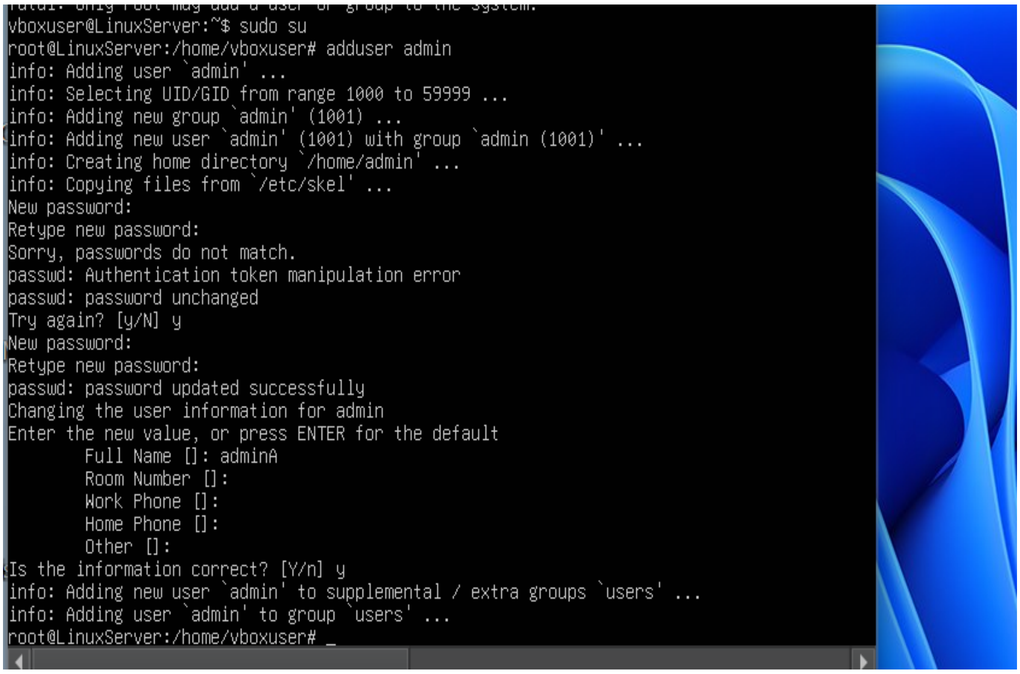
Enabled **unattended-upgrades** for automatic installation of stable updates.



```
root@LinuxServer:/home# cat /etc/apt/apt.conf.d/20auto-upgrades
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Unattended-Upgrade "1";
root@LinuxServer:/home#
```

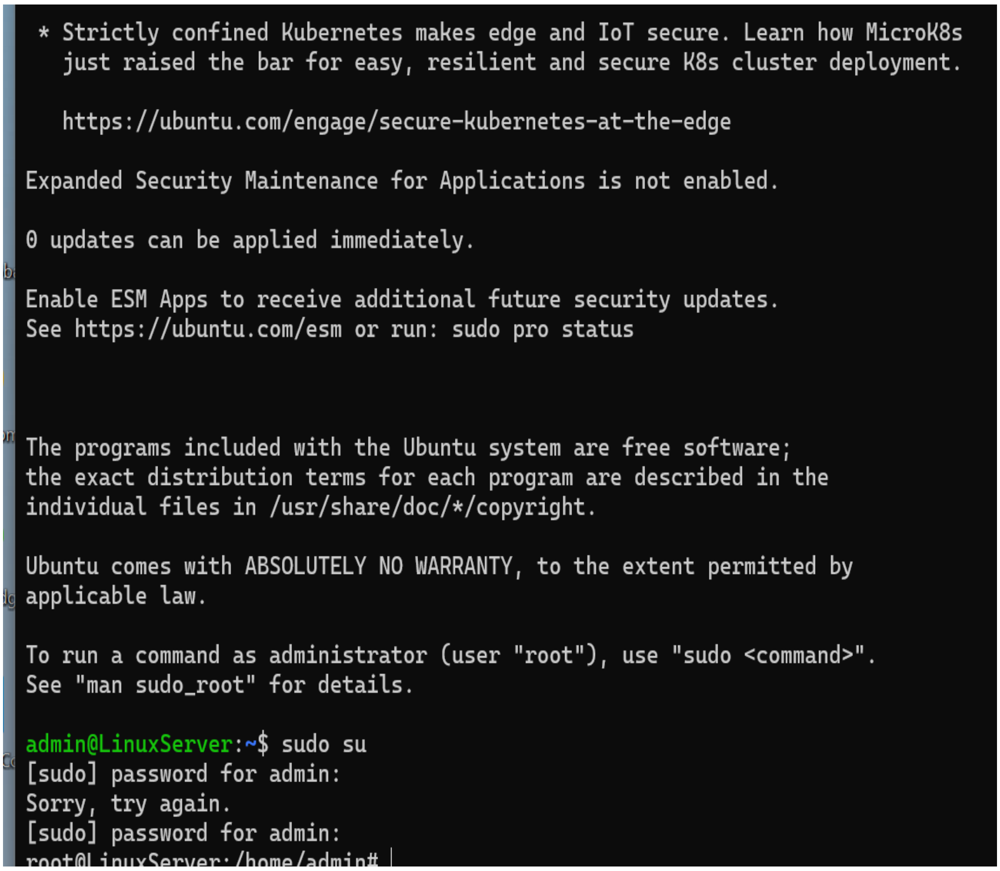
User & Privilege Management

Created a non-root user (admin) with „sudo” privileges.



```
fatal: Only root may add a user or group to the system.
vboxuser@LinuxServer:~$ sudo su
root@LinuxServer:/home/vboxuser# adduser admin
info: Adding user `admin' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `admin' (1001) ...
info: Adding new user `admin' (1001) with group `admin (1001)' ...
info: Creating home directory `/home/admin' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for admin
Enter the new value, or press ENTER for the default
        Full Name []: adminA
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `admin' to supplemental / extra groups `users' ...
info: Adding user `admin' to group `users' ...
root@LinuxServer:/home/vboxuser# _
```

Root login is restricted via SSH, ensuring administrative access requires an additional security step.

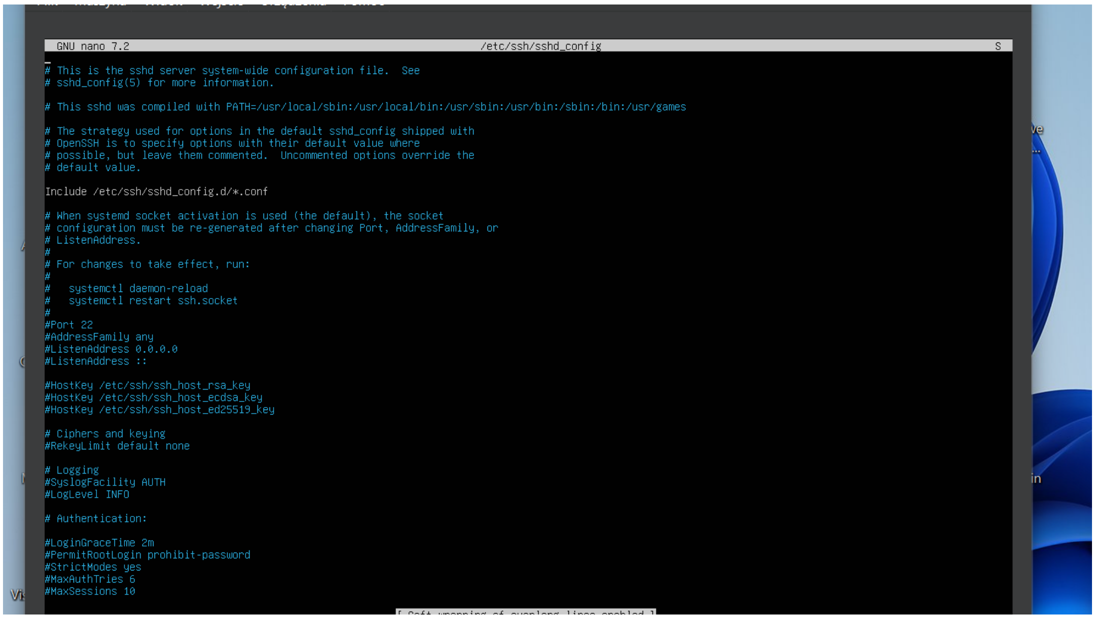


```
 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@LinuxServer:~$ sudo su
[sudo] password for admin:
Sorry, try again.
[sudo] password for admin:
root@LinuxServer:/home/admin# 
```

SSH Configuration

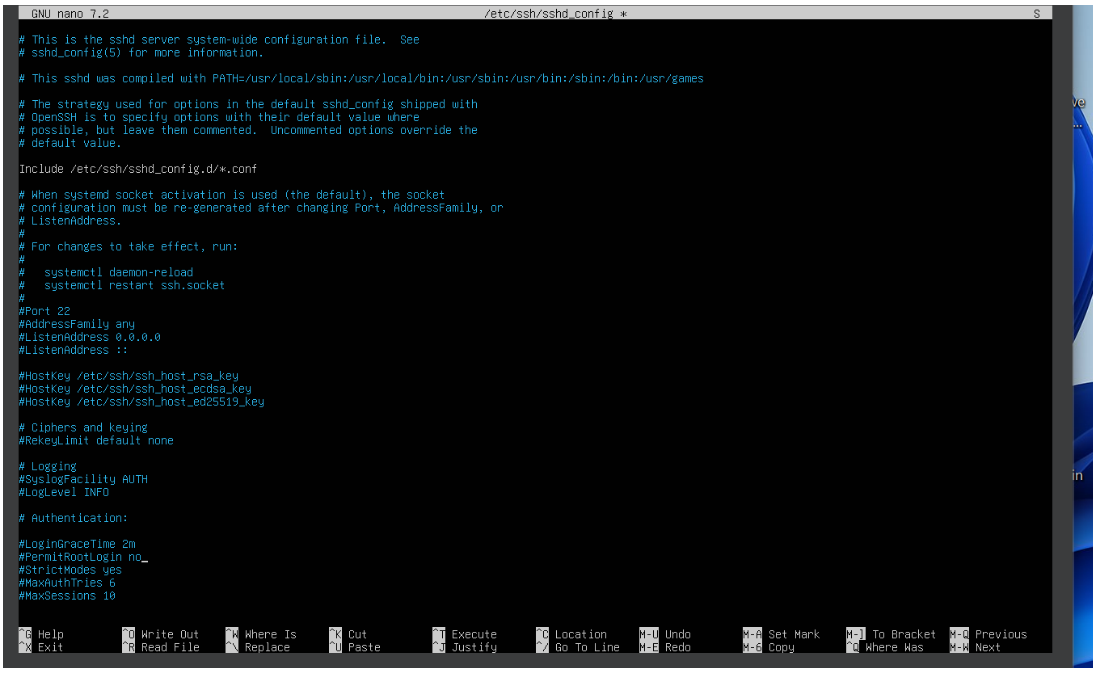
Disabled direct **root login** in /etc/ssh/sshd_config.

Default configuration:



```
  GNU nano 7.2                          /etc/ssh/sshd_config                           S

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/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

# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
#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

# Authentication:

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

                              [ Soft wrapping of overlong lines enabled ]
```

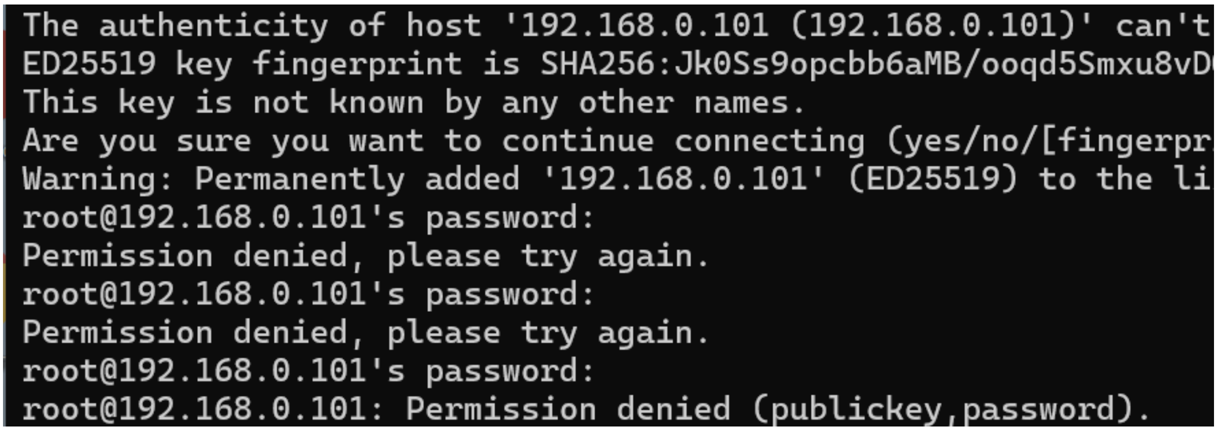
Add a change to the
PermitRootLogin prohibit-password line to PermitRootLogin no



```
  GNU nano 7.2                          /etc/ssh/sshd_config *                         S
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/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

# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
#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

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin no_
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was   M-W Next
```

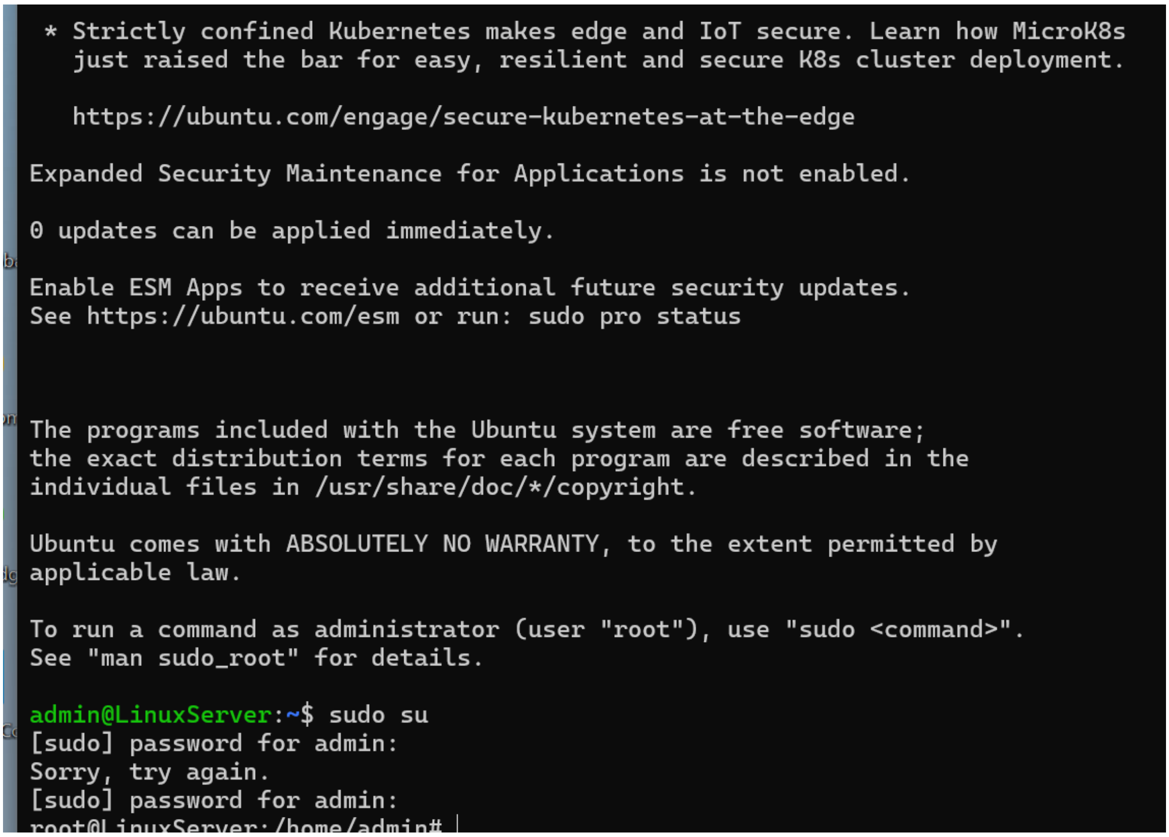
In the sshd_config file, the option PermitRootLogin prohibit-password means that the root user cannot log in with a password, but can still authenticate using an SSH key.

Changing this setting to no completely disables root login via SSH, which is more secure because it forces the use of a regular user with sudo privileges and eliminates a common brute-force attack vector against the root account.



```
The authenticity of host '192.168.0.101 (192.168.0.101)' can't
ED25519 key fingerprint is SHA256:Jk0Ss9opcbb6aMB/ooqd5Smxu8vD
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerpr
Warning: Permanently added '192.168.0.101' (ED25519) to the li
root@192.168.0.101's password:
Permission denied, please try again.
root@192.168.0.101's password:
Permission denied, please try again.
root@192.168.0.101's password:
root@192.168.0.101: Permission denied (publickey,password).
```

The screenshot below confirms that login is only possible via a regular user (admin) with sudo privileges.”



```
 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@LinuxServer:~$ sudo su
[sudo] password for admin:
Sorry, try again.
[sudo] password for admin:
root@LinuxServer:/home/admin#
```

Firewall (UFW)

The Uncomplicated Firewall (UFW) was enabled and configured to restrict incoming traffic.

Default policies were set to deny all incoming connections and allow all outgoing connections.

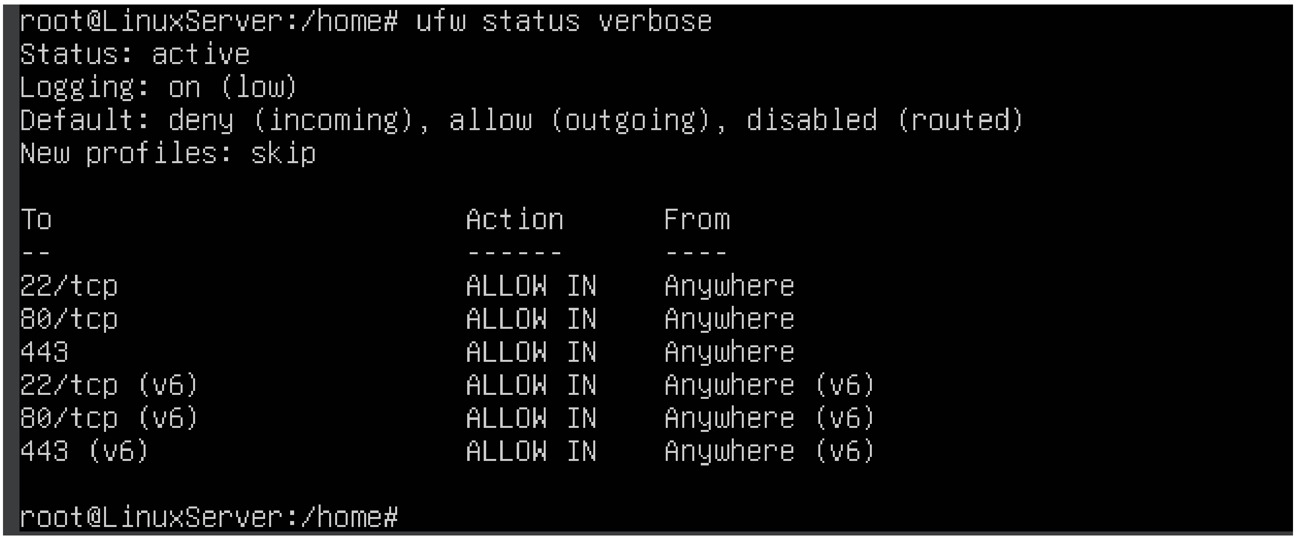
Only essential services were explicitly allowed:

SSH (22) – for remote administration,

HTTP (80) and HTTPS (443) – for web services.

The firewall rules appear in both IPv4 and IPv6 versions, which is why the status output shows six entries instead of three. This ensures that connections are properly controlled regardless of whether the client connects over IPv4 (e.g., 192.168.x.x) or IPv6 (e.g., fe80::...).

This configuration minimizes the server's attack surface by exposing only the required ports and blocking all unnecessary network traffic.



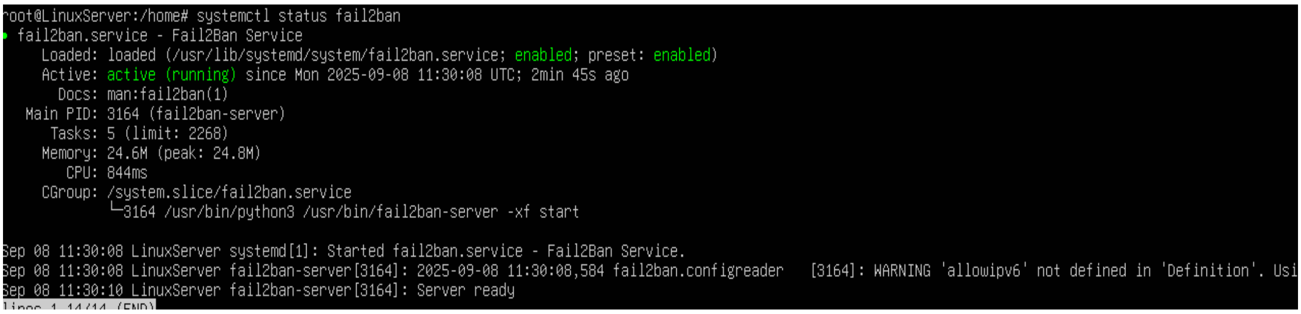
```
root@LinuxServer:/home# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
443                        ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443 (v6)                   ALLOW IN    Anywhere (v6)

root@LinuxServer:/home#
```

Fail2ban (Intrusion Prevention)

Fail2ban was installed and configured to monitor SSH login attempts and block IP addresses that exhibit suspicious behavior such as repeated failed logins.

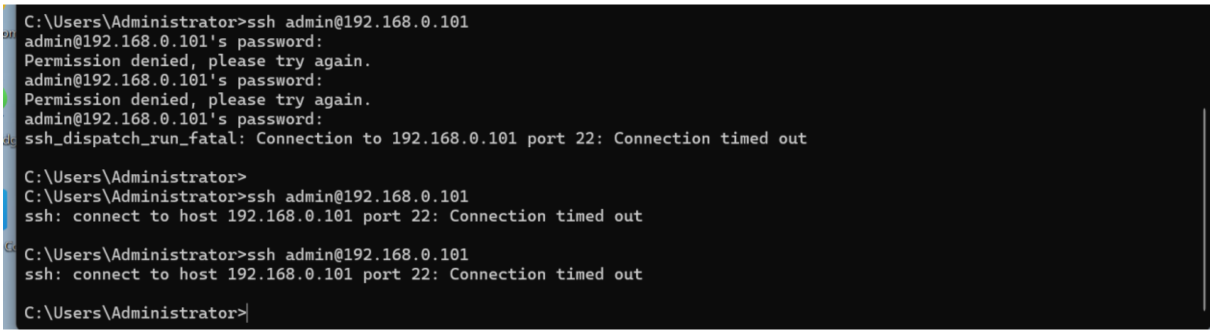
When multiple incorrect passwords were entered from the same client, Fail2ban automatically created a temporary firewall rule to block that IP address.

This significantly reduces the risk of brute-force attacks by preventing attackers from endlessly guessing credentials.



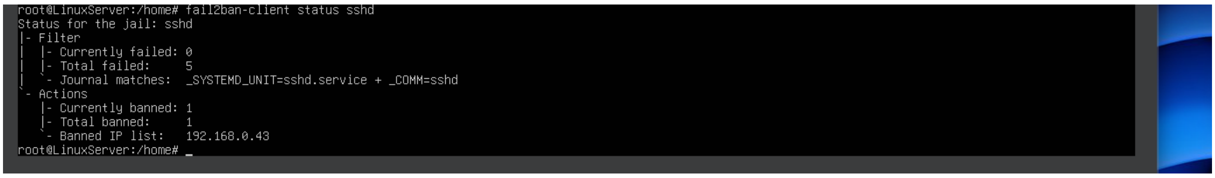
```
root@LinuxServer:/home# systemctl status fail2ban
• fail2ban.service - Fail2Ban Service
     Loaded: loaded (/usr/lib/systemd/system/fail2ban.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-09-08 11:30:08 UTC; 2min 45s ago
       Docs: man:fail2ban(1)
   Main PID: 3164 (fail2ban-server)
      Tasks: 5 (limit: 2268)
     Memory: 24.6M (peak: 24.8M)
        CPU: 844ms
     CGroup: /system.slice/fail2ban.service
             └─3164 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Sep 08 11:30:08 LinuxServer systemd[1]: Started fail2ban.service - Fail2Ban Service.
Sep 08 11:30:08 LinuxServer fail2ban-server[3164]: 2025-09-08 11:30:08,584 fail2ban.configreader   [3164]: WARNING 'allowipv6' not defined in 'Definition'. Usi
Sep 08 11:30:10 LinuxServer fail2ban-server[3164]: Server ready
lines 1-14/14 (END)
```

Below is an attempt to log in via ssh with an incorrect password and the result.



```
C:\Users\Administrator>ssh admin@192.168.0.101
admin@192.168.0.101's password:
Permission denied, please try again.
admin@192.168.0.101's password:
Permission denied, please try again.
admin@192.168.0.101's password:
ssh_dispatch_run_fatal: Connection to 192.168.0.101 port 22: Connection timed out

C:\Users\Administrator>
C:\Users\Administrator>ssh admin@192.168.0.101
ssh: connect to host 192.168.0.101 port 22: Connection timed out

C:\Users\Administrator>ssh admin@192.168.0.101
ssh: connect to host 192.168.0.101 port 22: Connection timed out

C:\Users\Administrator>
```

The IP address that performed repeated failed login attempts was automatically banned by Fail2ban and logged as shown below.



```
root@LinuxServer:/home# fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     5
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 1
   |- Total banned:     1
   `- Banned IP list:   192.168.0.43
root@LinuxServer:/home# _
```

Conclusion

The basic hardening of the Ubuntu server was successfully completed and verified through testing.

Key measures such as disabling direct root login, enforcing the use of a non-root user with sudo privileges, configuring a restrictive firewall, enabling automatic security updates, and deploying Fail2ban significantly reduced the attack surface.

These steps ensure that the server is more resilient against common threats such as brute-force attacks, unauthorized access, and outdated software vulnerabilities.

Author: Maciej Łęczycki

Date: 08.09.2025