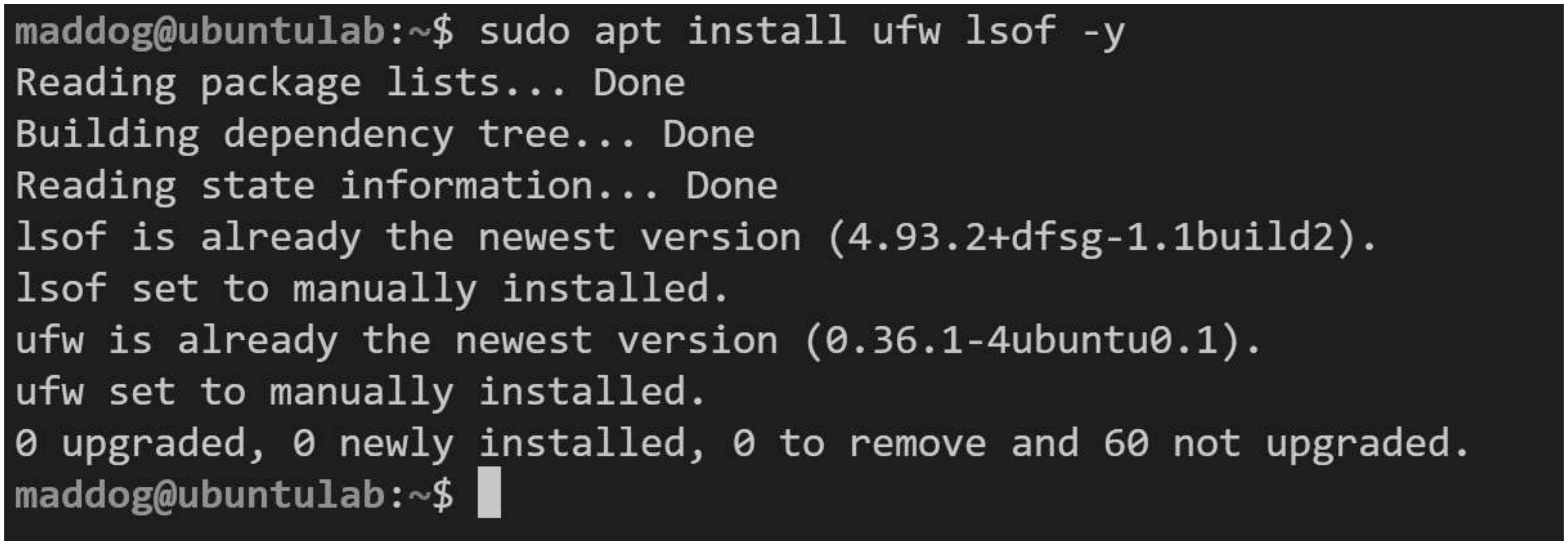
# UFW Firewall Configuration Lab

This document demonstrates the configuration and testing of Ubuntu's Uncomplicated Firewall (UFW). I describe a guide to

- Install and enable UFW with safe defaults.
- Add specific allow/deny rules (SSH, web, IP-based).
- Enable and tune logging, generate traffic, and read/interpret logs

## 1. Install UFW

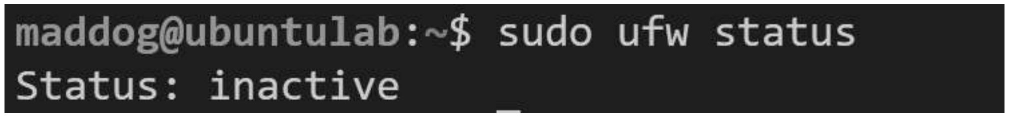


```
maddog@ubuntulab:~$ sudo apt install ufw lsof -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
lsof is already the newest version (4.93.2+dfsg-1.1build2).
lsof set to manually installed.
ufw is already the newest version (0.36.1-4ubuntu0.1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
maddog@ubuntulab:~$
```

### Overview:

The Uncomplicated Firewall (UFW) package is installed using `sudo apt install ufw`. UFW provides a simplified interface for managing iptables firewall rules on Ubuntu systems.

## 2. Check Initial Firewall Status

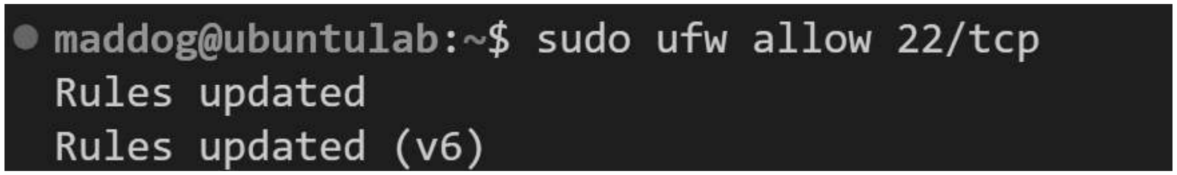


```
maddog@ubuntulab:~$ sudo ufw status
Status: inactive
```

### Overview:

Checking the initial status of UFW using `sudo ufw status`. By default, UFW is installed but inactive on fresh Ubuntu installations, meaning no firewall rules are being enforced.

## 3. Allow SSH to Prevent Lockout

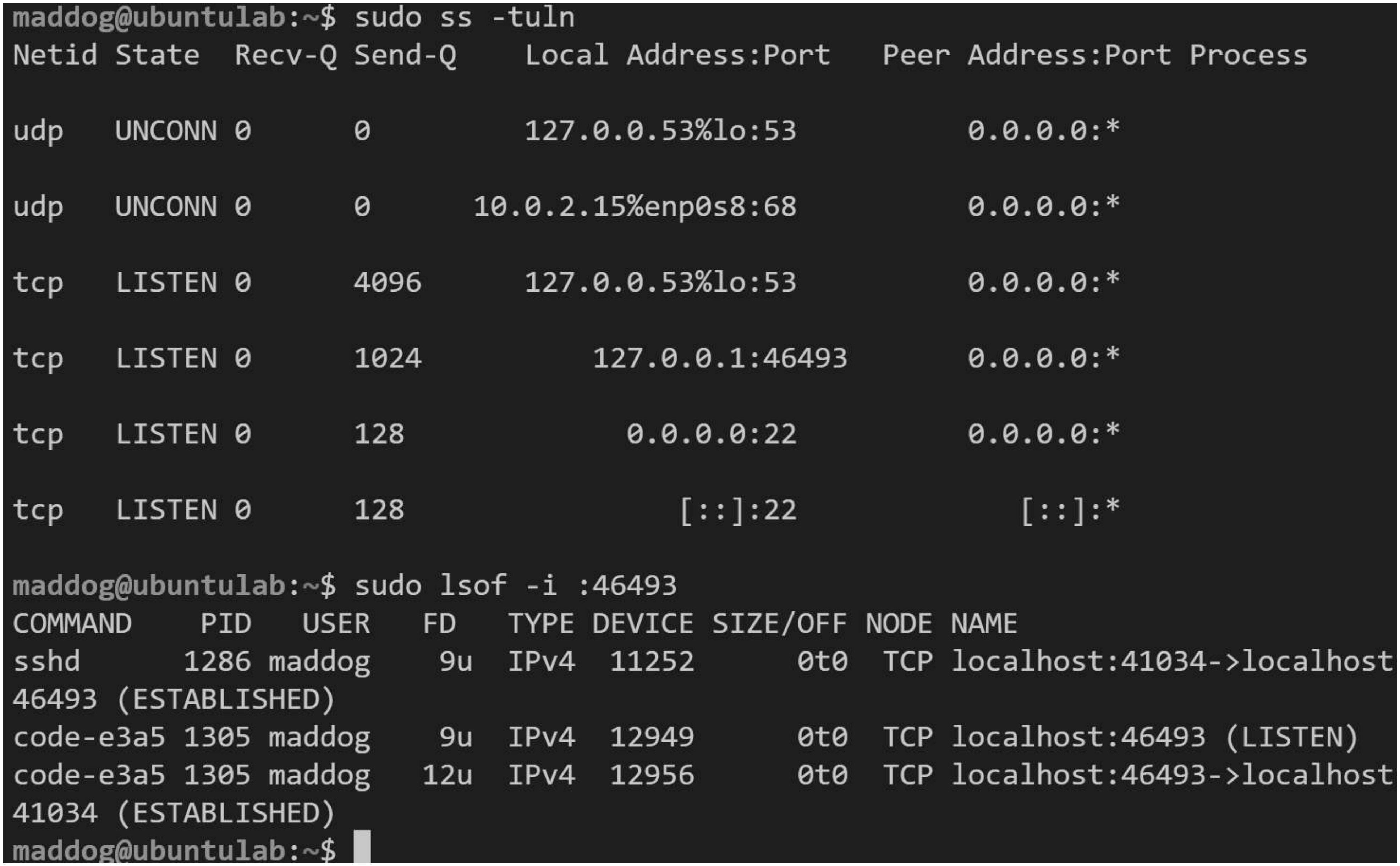


```
● maddog@ubuntulab:~$ sudo ufw allow 22/tcp
  Rules updated
  Rules updated (v6)
```

### Overview:

Before enabling the firewall, it’s crucial to allow SSH connections using `sudo ufw allow ssh` or `sudo ufw allow 22`. This prevents getting locked out of remote systems when the firewall is activated.

## 4. Investigate Listening Ports

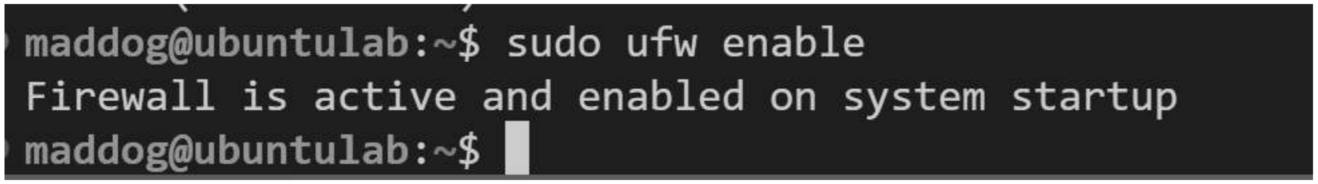


```
maddog@ubuntulab:~$ sudo ss -tuln
Netid State  Recv-Q Send-Q    Local Address:Port   Peer Address:Port Process

udp   UNCONN 0      0         127.0.0.53%lo:53          0.0.0.0:*

udp   UNCONN 0      0      10.0.2.15%enp0s8:68          0.0.0.0:*

tcp   LISTEN 0      4096      127.0.0.53%lo:53          0.0.0.0:*

tcp   LISTEN 0      1024          127.0.0.1:46493       0.0.0.0:*

tcp   LISTEN 0      128             0.0.0.0:22          0.0.0.0:*

tcp   LISTEN 0      128                [::]:22             [::]:*

maddog@ubuntulab:~$ sudo lsof -i :46493
COMMAND    PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
sshd      1286 maddog    9u  IPv4  11252      0t0  TCP localhost:41034->localhost
46493 (ESTABLISHED)
code-e3a5 1305 maddog    9u  IPv4  12949      0t0  TCP localhost:46493 (LISTEN)
code-e3a5 1305 maddog   12u  IPv4  12956      0t0  TCP localhost:46493->localhost
41034 (ESTABLISHED)
maddog@ubuntulab:~$
```

**Overview:**

Using `netstat -tulnp` or `ss -tulnp` to identify which services are currently listening on network ports. This helps determine which ports need firewall rules before enabling UFW.

## 5. Enable UFW

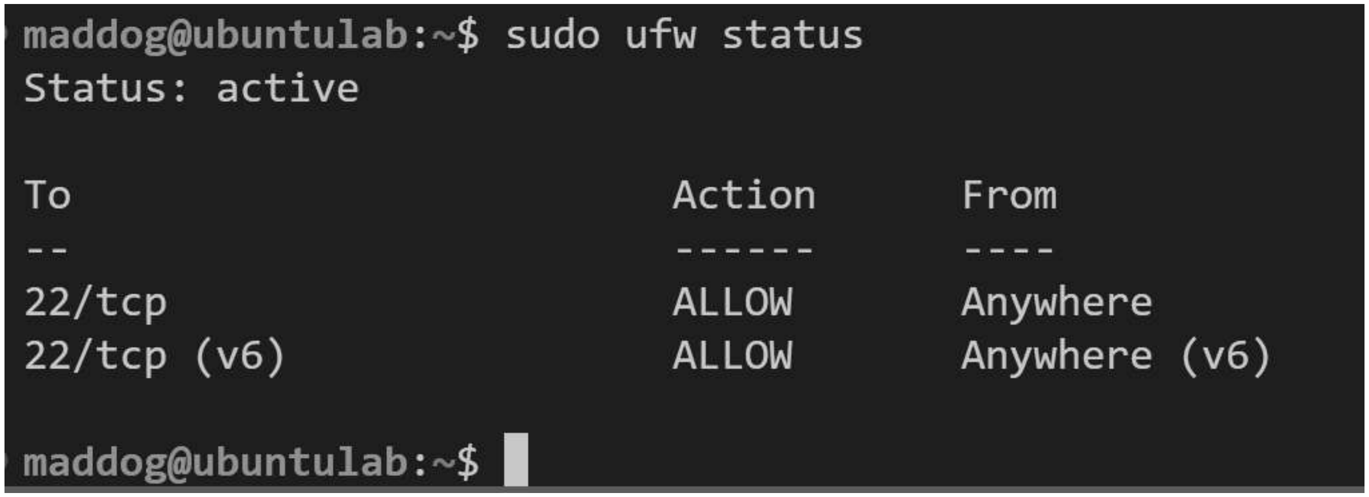


```
maddog@ubuntulab:~$ sudo ufw enable
Firewall is active and enabled on system startup
maddog@ubuntulab:~$
```

**Overview:**

Activating the firewall using `sudo ufw enable`. Once enabled, UFW will enforce the configured rules and block unauthorized connections while allowing previously defined exceptions.

## 6. Verify UFW Status After Enabling

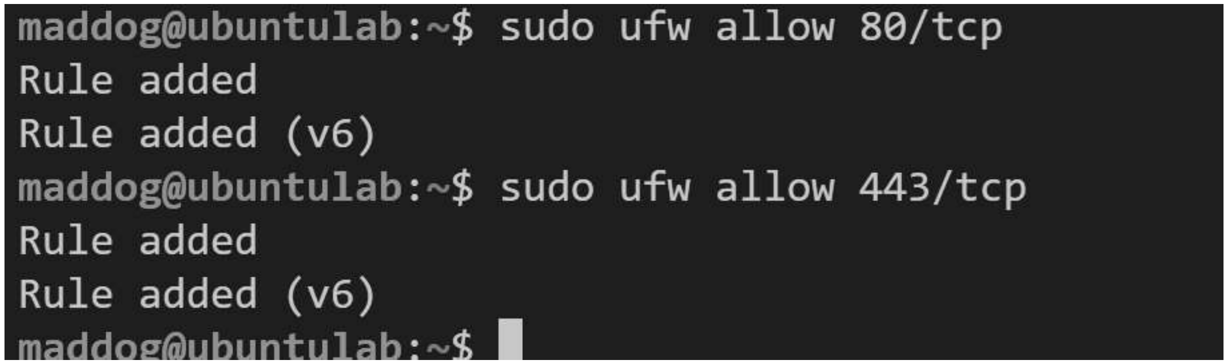


```
maddog@ubuntulab:~$ sudo ufw status
Status: active

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

maddog@ubuntulab:~$
```

**Overview:**

Confirming that UFW is now active and displaying the current rule set. The status shows that SSH (port 22) is allowed from anywhere, ensuring continued remote access.

## 7. Allow HTTP and HTTPS Traffic

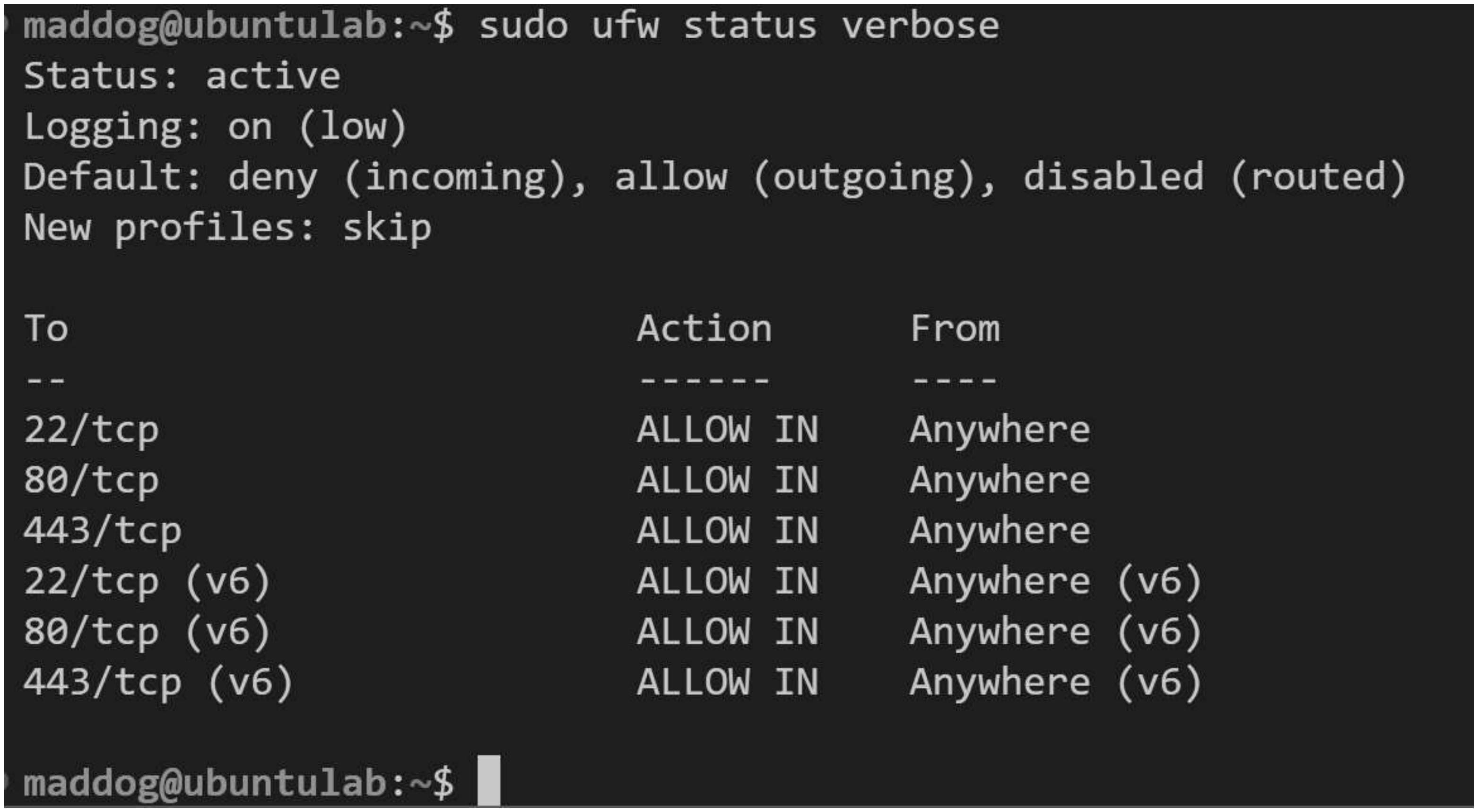


```
maddog@ubuntulab:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
maddog@ubuntulab:~$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
maddog@ubuntulab:~$ 
```

### Overview:

Adding rules to allow web traffic using `sudo ufw allow http` and `sudo ufw allow https`. This permits incoming connections on ports 80 and 443 for web server functionality.

## 8. Configure Outgoing and Incoming Traffic Policies

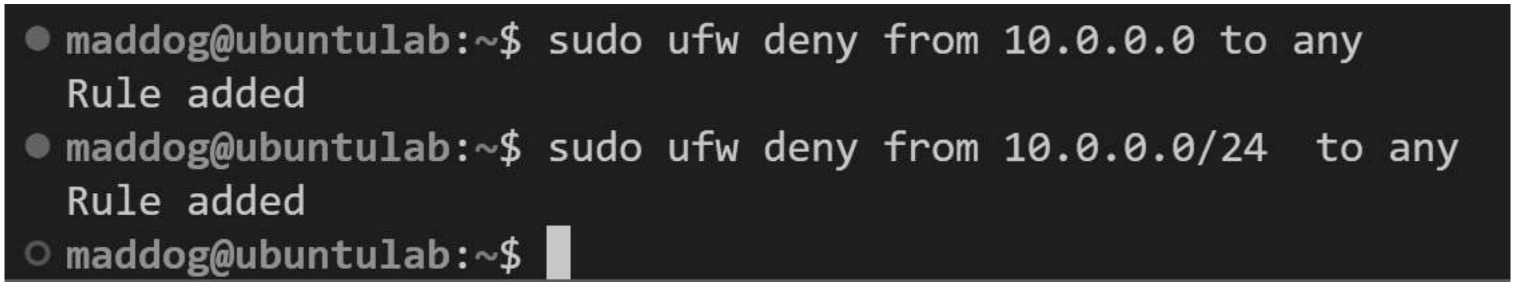


```
maddog@ubuntulab:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

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

maddog@ubuntulab:~$
```

**Overview:**

Setting default policies for traffic direction using `sudo ufw default deny incoming` and `sudo ufw default allow outgoing`. This creates a secure-by-default configuration that blocks unsolicited incoming connections while allowing outbound communication.

## 9. Blacklist IP Addresses and Subnets

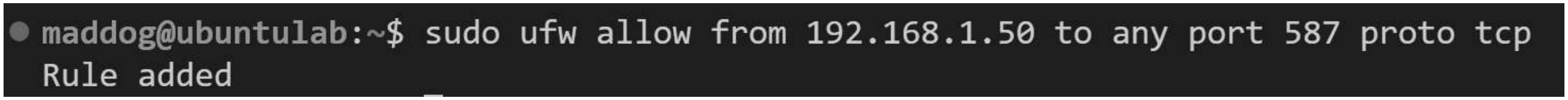


```
maddog@ubuntulab:~$ sudo ufw deny from 10.0.0.0 to any
Rule added
maddog@ubuntulab:~$ sudo ufw deny from 10.0.0.0/24  to any
Rule added
maddog@ubuntulab:~$
```

### Overview:

Demonstrating how to block specific IP addresses and entire subnets using `sudo ufw deny from [IP/subnet]`. This shows both single IP blocking and CIDR notation for subnet-wide restrictions.

## 10. Allow Specific Host to Specific Port

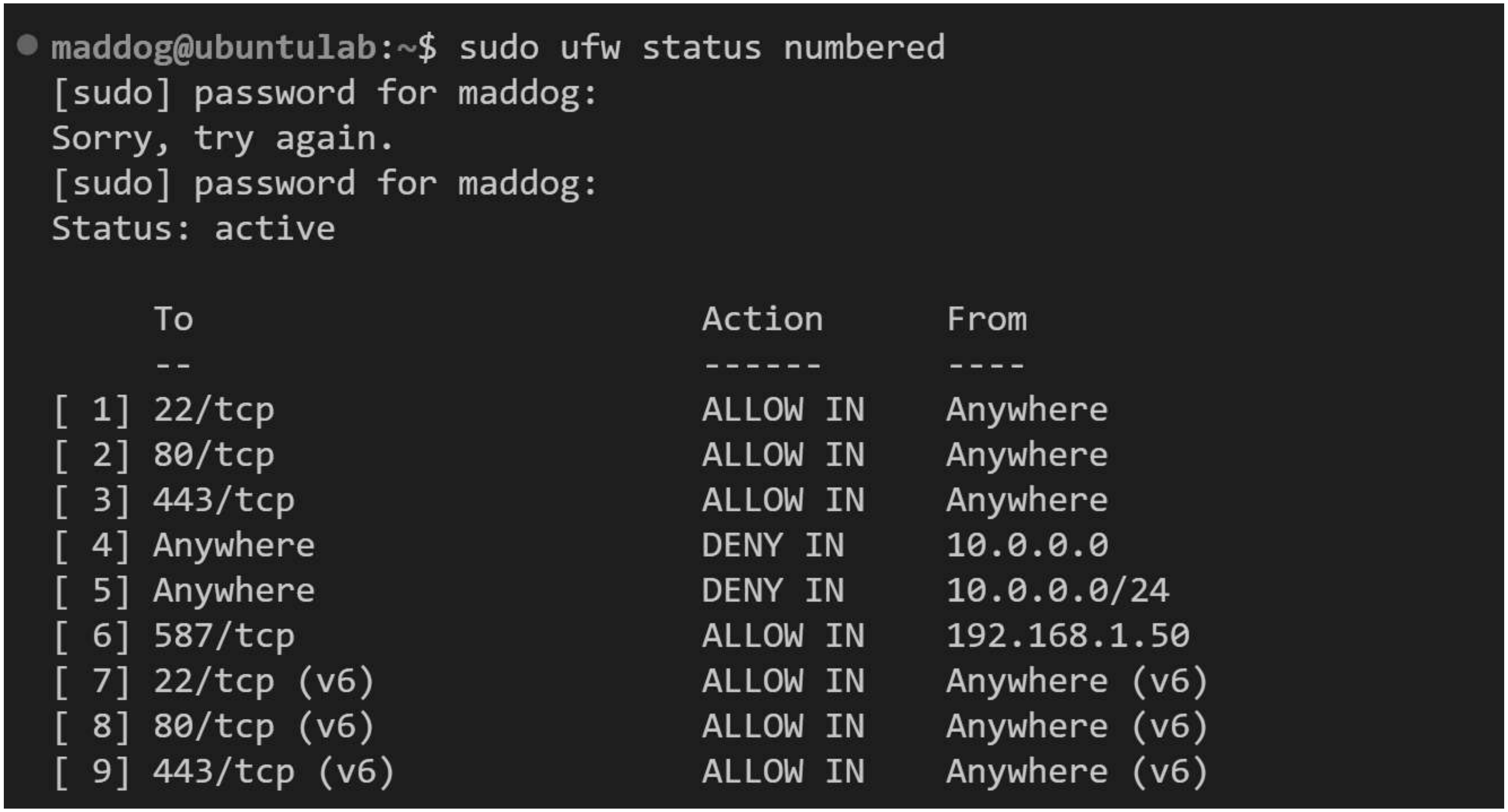


```
maddog@ubuntulab:~$ sudo ufw allow from 192.168.1.50 to any port 587 proto tcp
Rule added
```

### Overview:

Creating granular firewall rules that allow specific IP addresses to access particular ports. This demonstrates advanced rule creation for precise access control: `sudo ufw allow from [IP] to any port [PORT]`.

## 11. List All Firewall Rules

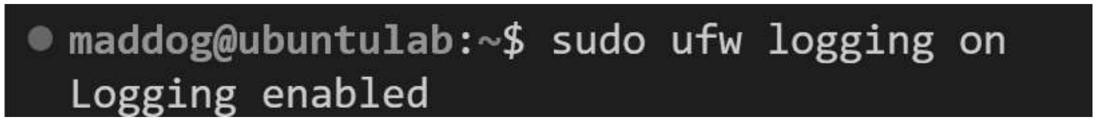


```
● maddog@ubuntulab:~$ sudo ufw status numbered
  [sudo] password for maddog:
  Sorry, try again.
  [sudo] password for maddog:
  Status: active

       To                         Action      From
       --                         ------      ----
  [ 1] 22/tcp                     ALLOW IN    Anywhere
  [ 2] 80/tcp                     ALLOW IN    Anywhere
  [ 3] 443/tcp                    ALLOW IN    Anywhere
  [ 4] Anywhere                   DENY IN     10.0.0.0
  [ 5] Anywhere                   DENY IN     10.0.0.0/24
  [ 6] 587/tcp                    ALLOW IN    192.168.1.50
  [ 7] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
  [ 8] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
  [ 9] 443/tcp (v6)               ALLOW IN    Anywhere (v6)
```

**Overview:**

Displaying all configured firewall rules using `sudo ufw status numbered` or `sudo ufw status verbose`. This provides a comprehensive view of all active rules with their priorities and detailed configurations.

## 12. Enable UFW Logging

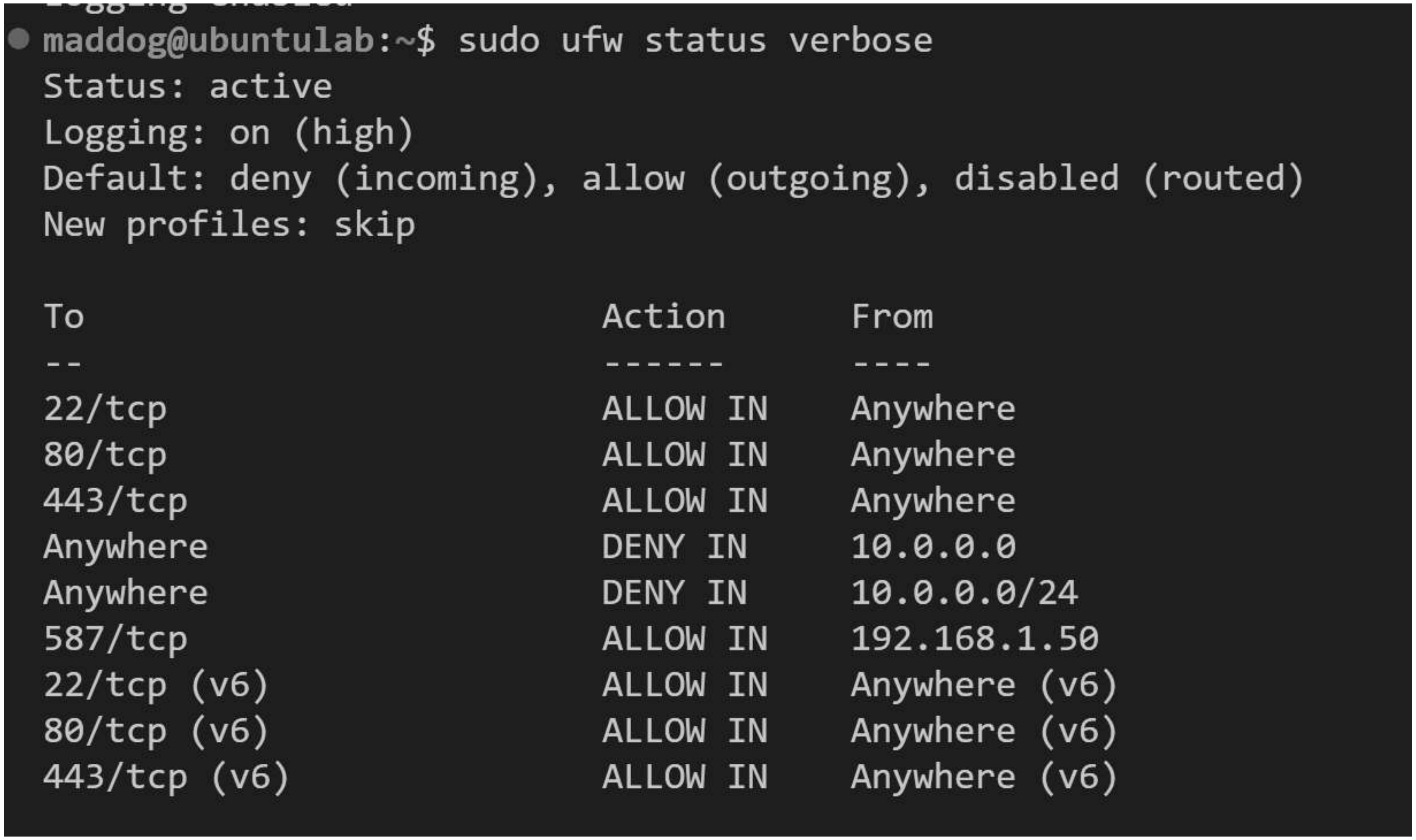


```
● maddog@ubuntulab:~$ sudo ufw logging on
  Logging enabled
```

**Overview:**

Activating UFW logging functionality using `sudo ufw logging on`. This enables the firewall to record blocked and allowed connections for security monitoring and troubleshooting purposes.

## 13. Set Logging Level to High

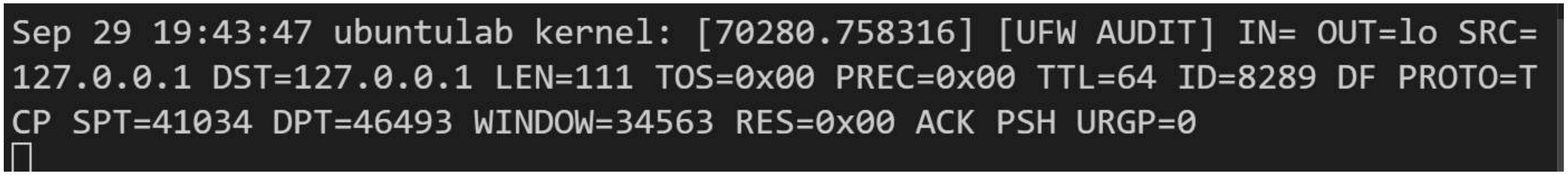


```
maddog@ubuntulab:~$ sudo ufw status verbose
Status: active
Logging: on (high)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
Anywhere                   DENY IN     10.0.0.0
Anywhere                   DENY IN     10.0.0.0/24
587/tcp                    ALLOW IN    192.168.1.50
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)
```

**Overview:**

Configuring verbose logging using `sudo ufw logging high`. Higher logging levels capture more detailed information about firewall activities, including all blocked packets and rule matches.

## 14. View Firewall Logs

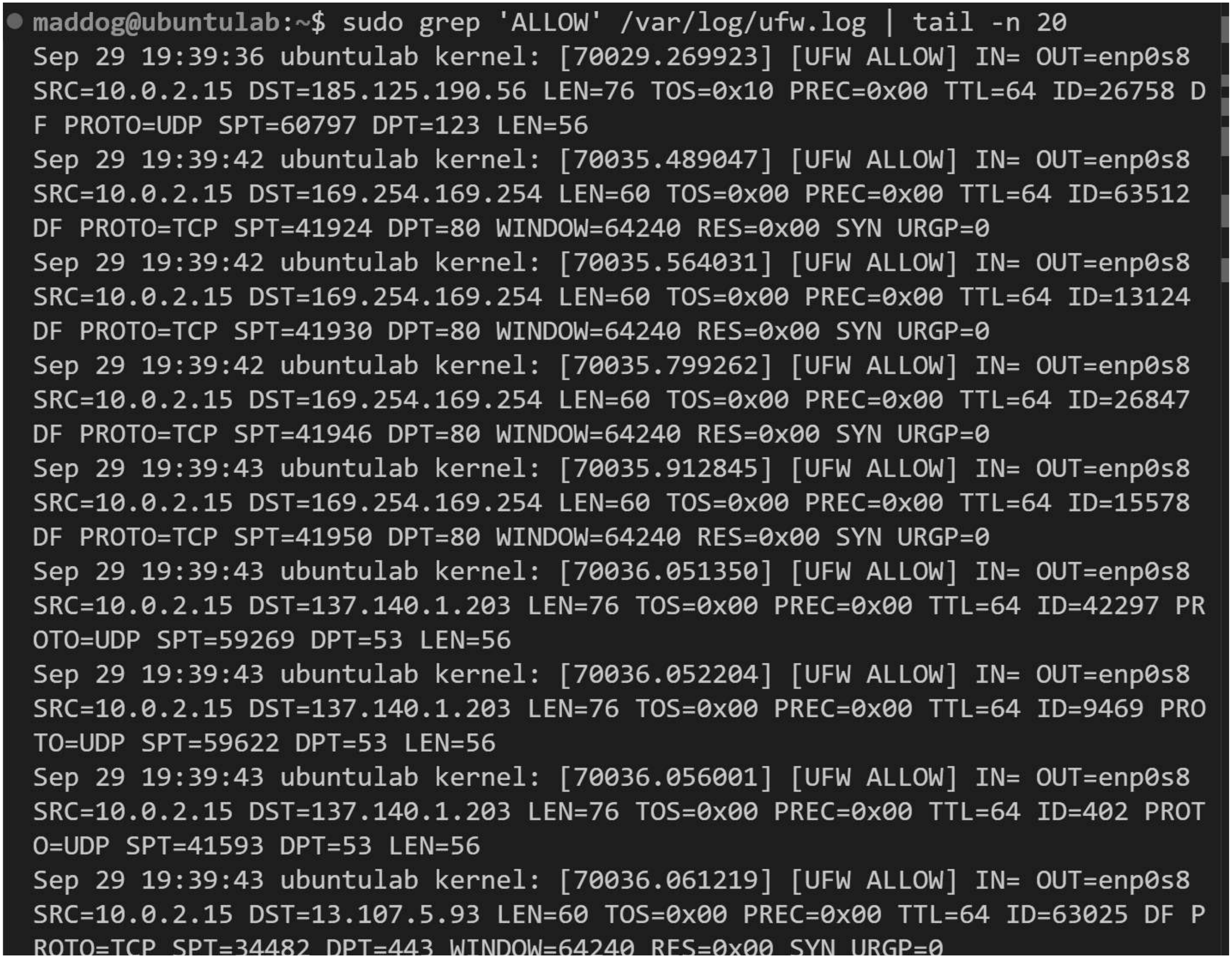


```
Sep 29 19:43:47 ubuntulab kernel: [70280.758316] [UFW AUDIT] IN= OUT=lo SRC=
127.0.0.1 DST=127.0.0.1 LEN=111 TOS=0x00 PREC=0x00 TTL=64 ID=8289 DF PROTO=T
CP SPT=41034 DPT=46493 WINDOW=34563 RES=0x00 ACK PSH URGP=0
```

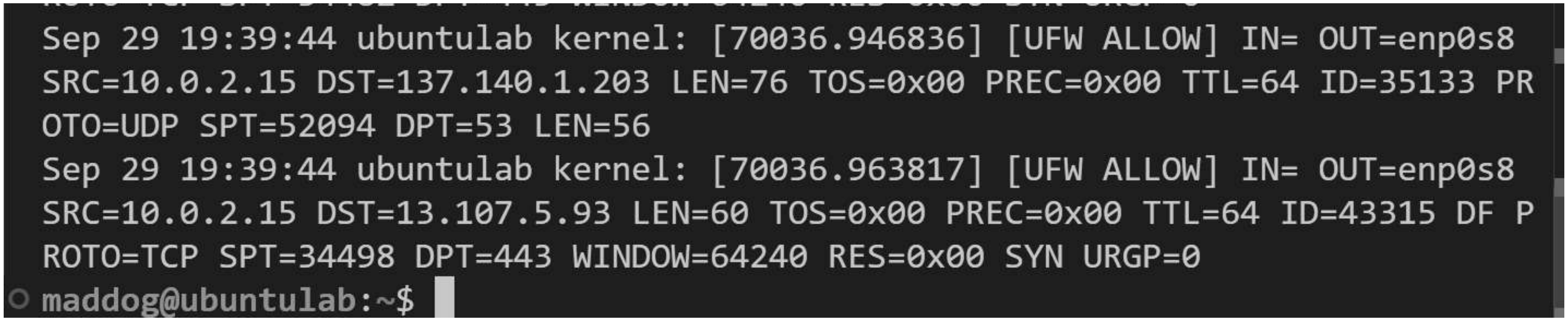
### Overview:

Monitoring firewall logs in real-time using `sudo tail -f /var/log/ufw.log`. This shows live firewall activity, including blocked connection attempts and rule matches.

## 15. Filter Log Entries



```
maddog@ubuntulab:~$ sudo grep 'ALLOW' /var/log/ufw.log | tail -n 20
Sep 29 19:39:36 ubuntulab kernel: [70029.269923] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=185.125.190.56 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=26758 D
F PROTO=UDP SPT=60797 DPT=123 LEN=56
Sep 29 19:39:42 ubuntulab kernel: [70035.489047] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=169.254.169.254 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=63512
DF PROTO=TCP SPT=41924 DPT=80 WINDOW=64240 RES=0x00 SYN URGP=0
Sep 29 19:39:42 ubuntulab kernel: [70035.564031] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=169.254.169.254 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=13124
DF PROTO=TCP SPT=41930 DPT=80 WINDOW=64240 RES=0x00 SYN URGP=0
Sep 29 19:39:42 ubuntulab kernel: [70035.799262] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=169.254.169.254 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=26847
DF PROTO=TCP SPT=41946 DPT=80 WINDOW=64240 RES=0x00 SYN URGP=0
Sep 29 19:39:43 ubuntulab kernel: [70035.912845] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=169.254.169.254 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=15578
DF PROTO=TCP SPT=41950 DPT=80 WINDOW=64240 RES=0x00 SYN URGP=0
Sep 29 19:39:43 ubuntulab kernel: [70036.051350] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=137.140.1.203 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=42297 PR
OTO=UDP SPT=59269 DPT=53 LEN=56
Sep 29 19:39:43 ubuntulab kernel: [70036.052204] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=137.140.1.203 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=9469 PRO
TO=UDP SPT=59622 DPT=53 LEN=56
Sep 29 19:39:43 ubuntulab kernel: [70036.056001] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=137.140.1.203 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=402 PROT
O=UDP SPT=41593 DPT=53 LEN=56
Sep 29 19:39:43 ubuntulab kernel: [70036.061219] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=13.107.5.93 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=63025 DF P
ROTO=TCP SPT=34482 DPT=443 WINDOW=64240 RES=0x00 SYN URGP=0
```



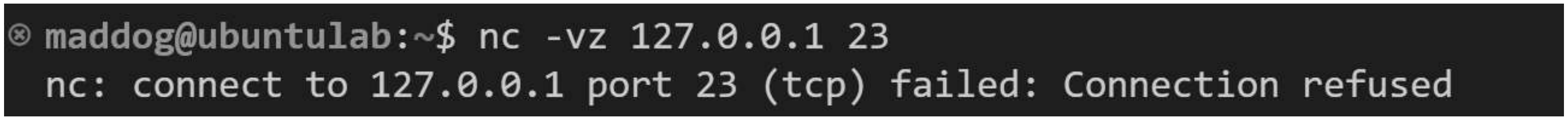
```
Sep 29 19:39:44 ubuntulab kernel: [70036.946836] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=137.140.1.203 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=35133 PR
OTO=UDP SPT=52094 DPT=53 LEN=56
Sep 29 19:39:44 ubuntulab kernel: [70036.963817] [UFW ALLOW] IN= OUT=enp0s8
SRC=10.0.2.15 DST=13.107.5.93 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=43315 DF P
ROTO=TCP SPT=34498 DPT=443 WINDOW=64240 RES=0x00 SYN URGP=0
maddog@ubuntulab:~$
```

**Overview:**

Using grep to filter specific log entries and analyze firewall behavior. This demonstrates how to search for particular events, IP addresses, or ports in the firewall logs for targeted analysis.

## Network Testing with Netcat

### Netcat Verbose Mode

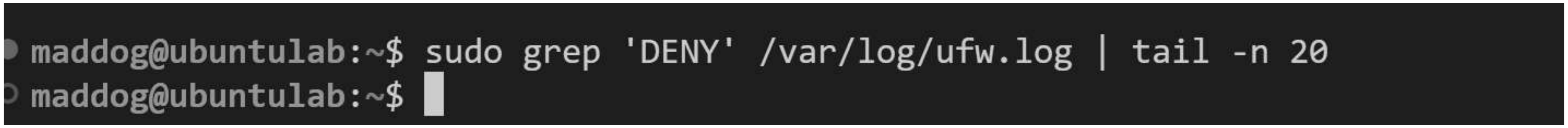


```
maddog@ubuntulab:~$ nc -vz 127.0.0.1 23
nc: connect to 127.0.0.1 port 23 (tcp) failed: Connection refused
```

**Overview:**

Using netcat (`nc`) in verbose mode to test network connectivity and firewall rules. The `-v` flag provides detailed output about connection attempts, while `-z` performs port scanning without sending data.

### Denied Connection Log

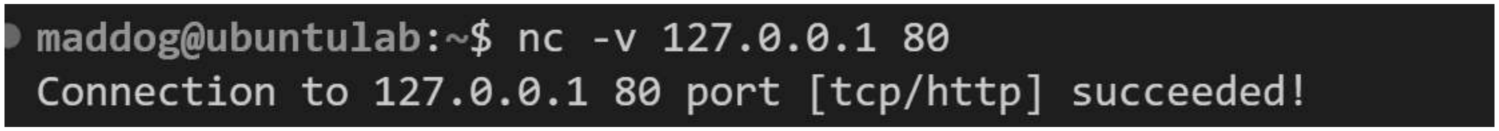


```
maddog@ubuntulab:~$ sudo grep 'DENY' /var/log/ufw.log | tail -n 20
maddog@ubuntulab:~$
```

**Overview:**

Log entry showing that no connections were denied during testing, indicating that the tested connections matched allow rules or were directed to permitted services.

### Allowed Connection Log

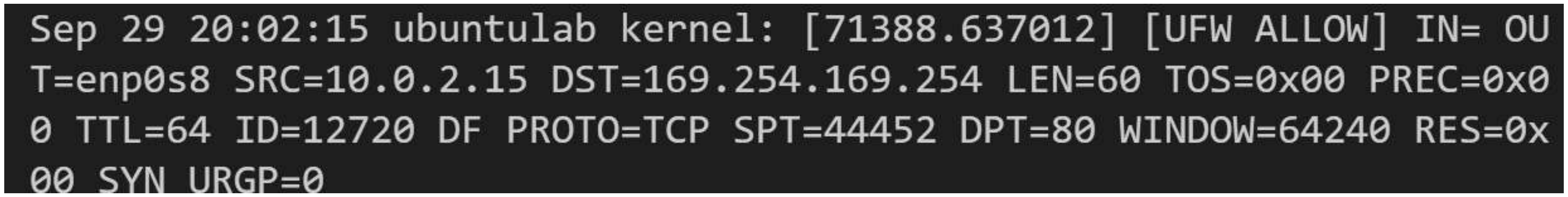


```
maddog@ubuntulab:~$ nc -v 127.0.0.1 80
Connection to 127.0.0.1 80 port [tcp/http] succeeded!
```

### Overview:

Log entry demonstrating a successful connection that was permitted by the firewall rules, showing how UFW logs both allowed and denied traffic when logging is enabled.

### Specific IP Allow Log



```
Sep 29 20:02:15 ubuntulab kernel: [71388.637012] [UFW ALLOW] IN= OU
T=enp0s8 SRC=10.0.2.15 DST=169.254.169.254 LEN=60 TOS=0x00 PREC=0x0
0 TTL=64 ID=12720 DF PROTO=TCP SPT=44452 DPT=80 WINDOW=64240 RES=0x
00 SYN URGP=0
```

### Overview:

Detailed log entry showing an allowed connection from IP address 10.0.2.15, demonstrating how UFW logs include source IP information and which rule permitted the connection.

## Summary

This lab demonstrates comprehensive UFW firewall configuration including:

- Installation and initial setup
- Rule creation for common services (SSH, HTTP, HTTPS)
- Advanced rule configuration for specific hosts and ports
- IP and subnet blacklisting
- Logging configuration and analysis
- Network testing and validation

UFW provides an effective, user-friendly interface for managing Linux firewall rules while maintaining the power and flexibility of the underlying iptables system.