

# Domain Blocker Task

Yehia A. Mostafa

The Link to the [Github Repo](#)

## Blocking a Domain Using **iptables** and **ip6tables** (Step-by-Step with Explanations)

### Step-by-Step Domain Blocking Script

#### 1. Create the Script File

```
nano DNStask.sh
```

Paste the following content:

```
#!/bin/bash
read -p "Enter Domain Name: " domain
for ip4 in $(dig +short "$domain" A); do
    echo "Blocking IPv4: $ip4"
    sudo iptables -A OUTPUT -d "$ip4" -j DROP
done
for ip6 in $(dig +short "$domain" AAAA); do
    echo "Blocking IPv6: $ip6"
    sudo ip6tables -A OUTPUT -d "$ip6" -j DROP
done

sudo iptables-save > /etc/rules.v4
sudo ip6tables-save > /etc/rules.v6

echo "Disallowed forwarding traffic to $domain"
cat /etc/rules.v6
cat /etc/rules.v4
```

## 2. Make the Script Executable

```
chmod +x DNStask.sh
```

## Explanation of the Script Components

**read -p "Enter Domain Name: " domain**

- Prompts the user to enter the domain name to block.

## Resolving and Blocking IPv4 Addresses

```
for ip4 in $(dig +short "$domain" A); do
    echo "Blocking IPv4: $ip4"
    sudo iptables -A OUTPUT -d "$ip4" -j DROP
done
```

- **dig +short "\$domain" A** : Resolves all **IPv4** addresses of the domain.
- **iptables -A OUTPUT -d "\$ip4" -j DROP** : Blocks outgoing traffic to each resolved IPv4 address.
  - **A OUTPUT** : Appends the rule to the OUTPUT chain (used for traffic from this machine).
  - **d "\$ip4"** : Specifies the destination IP address.
  - **j DROP** : Silently drops matching packets.

## Resolving and Blocking IPv6 Addresses

```
for ip6 in $(dig +short "$domain" AAAA); do
    echo "Blocking IPv6: $ip6"
    sudo ip6tables -A OUTPUT -d "$ip6" -j DROP
done
```

- **dig +short "\$domain" AAAA** : Resolves all **IPv6** addresses of the domain.
- **ip6tables -A OUTPUT -d "\$ip6" -j DROP** : Blocks outgoing IPv6 traffic to each resolved address.

## Saving the Firewall Rules

```
sudo iptables-save > /etc/rules.v4
sudo ip6tables-save > /etc/rules.v6
```

- Saves the current IPv4 and IPv6 rules to files.
- These files can be reloaded later or used with `iptables-persistent`.

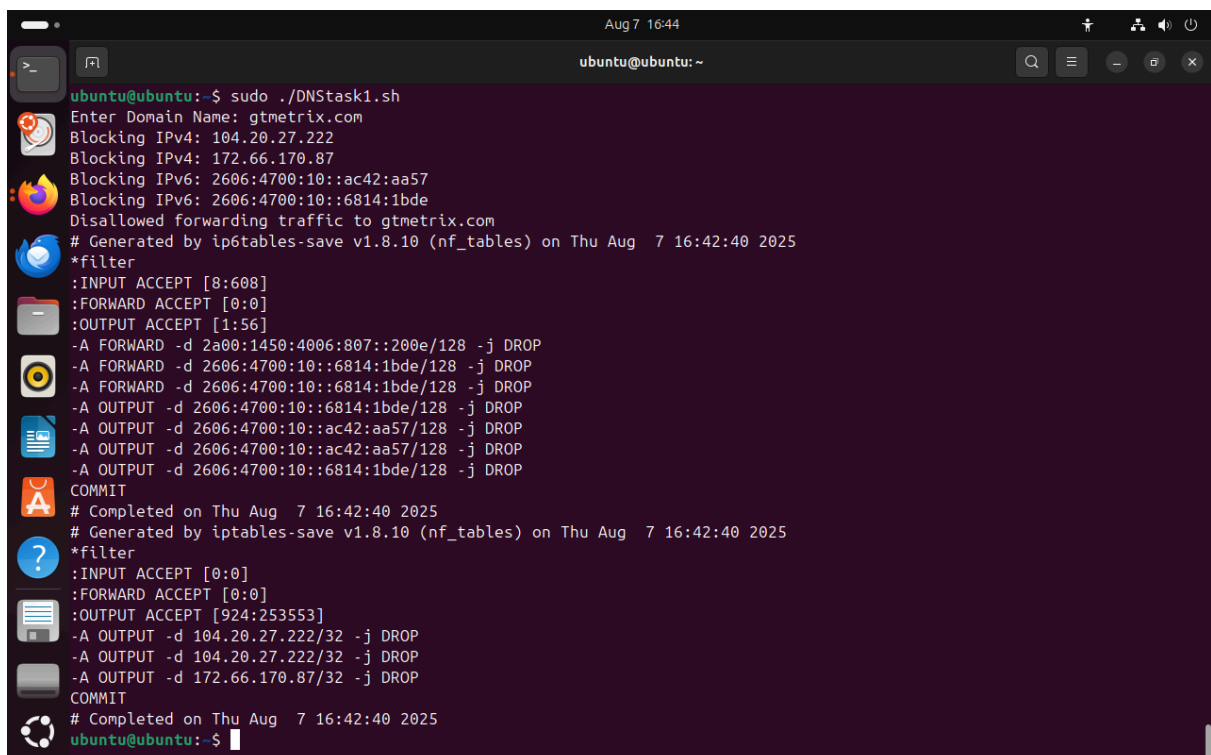
## Verifying the Rules

```
cat /etc/rules.v4
cat /etc/rules.v6
```

## Running the Script

```
sudo ./DNStask.sh
```

### Output:



```
ubuntu@ubuntu:~$ sudo ./DNStask1.sh
Enter Domain Name: gtmetrix.com
Blocking IPv4: 104.20.27.222
Blocking IPv4: 172.66.170.87
Blocking IPv6: 2606:4700:10::ac42:aa57
Blocking IPv6: 2606:4700:10::6814:1bde
Disallowed forwarding traffic to gtmetrix.com
# Generated by ip6tables-save v1.8.10 (nf_tables) on Thu Aug  7 16:42:40 2025
*filter
:INPUT ACCEPT [8:608]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1:56]
-A FORWARD -d 2a00:1450:4006:807::200e/128 -j DROP
-A FORWARD -d 2606:4700:10::6814:1bde/128 -j DROP
-A FORWARD -d 2606:4700:10::6814:1bde/128 -j DROP
-A OUTPUT -d 2606:4700:10::6814:1bde/128 -j DROP
-A OUTPUT -d 2606:4700:10::ac42:aa57/128 -j DROP
-A OUTPUT -d 2606:4700:10::ac42:aa57/128 -j DROP
-A OUTPUT -d 2606:4700:10::6814:1bde/128 -j DROP
COMMIT
# Completed on Thu Aug  7 16:42:40 2025
# Generated by iptables-save v1.8.10 (nf_tables) on Thu Aug  7 16:42:40 2025
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [924:253553]
-A OUTPUT -d 104.20.27.222/32 -j DROP
-A OUTPUT -d 104.20.27.222/32 -j DROP
-A OUTPUT -d 172.66.170.87/32 -j DROP
COMMIT
# Completed on Thu Aug  7 16:42:40 2025
ubuntu@ubuntu:~$
```

## Verifying the Block

```
ubuntu@ubuntu:~$ ping gtmetrix.com
PING gtmetrix.com (104.20.27.222) 56(84) bytes of data.
^C
--- gtmetrix.com ping statistics ---
14 packets transmitted, 0 received, 100% packet loss, time 13330ms
```

you can find the testing video [here](#)

---

## Challenges that faced me

- I kept editing the script but visiting the website worked then I figured out that I had to do a for loop and get all IPs because some websites' DNS are hosted on multiple IPs for load balancing.