Firewall Setup and Configuration – Windows & Linux (Kali)

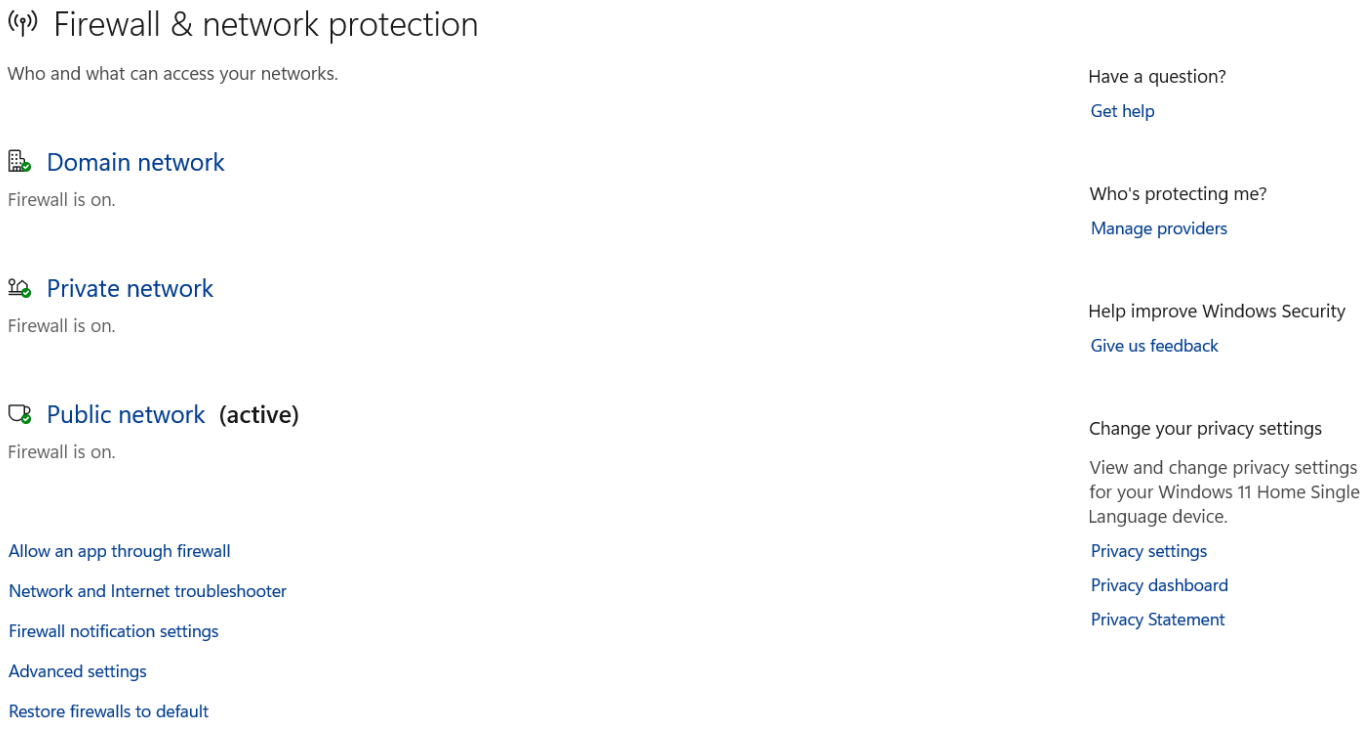
Step 1 Open Windows Firewall and capture a baseline

Using **GUI** -

1. Click **Start** > type **Windows Security** > open it.

2. Click **Firewall & network protection**.

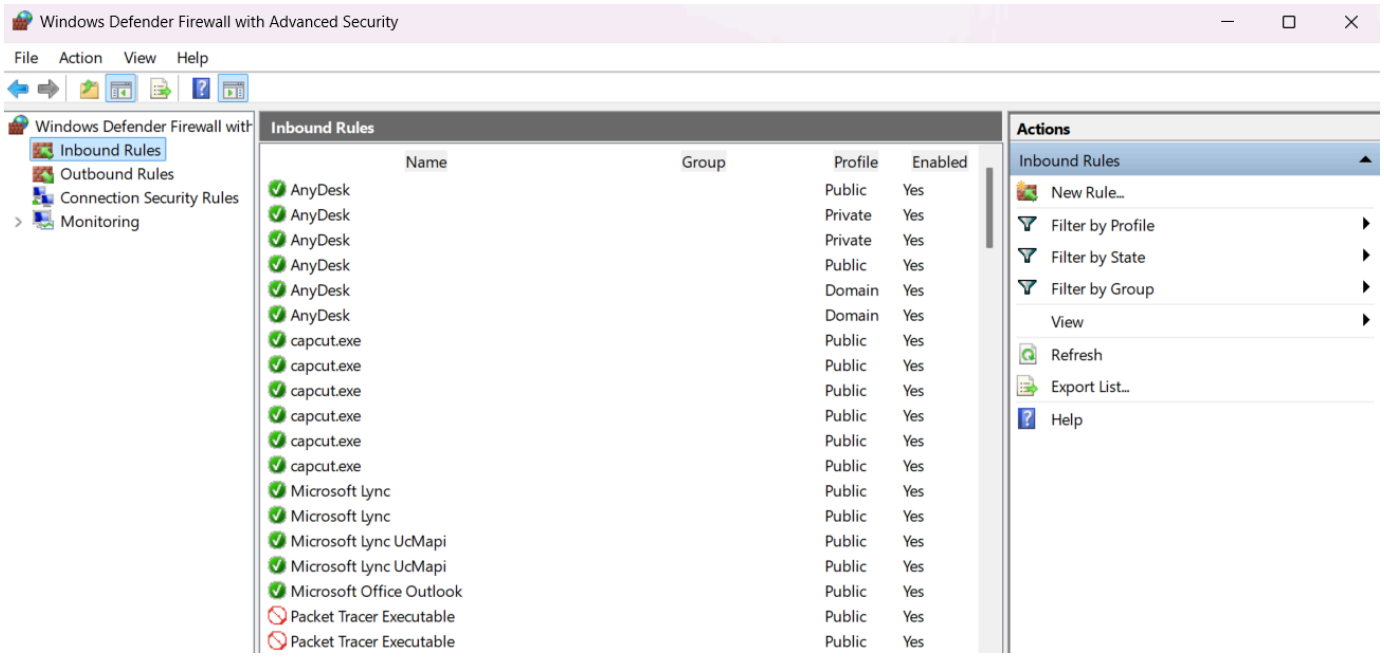
   Note which profile is *Active* (Domain / Private / Public) and whether the firewall is **On** or **Off**.



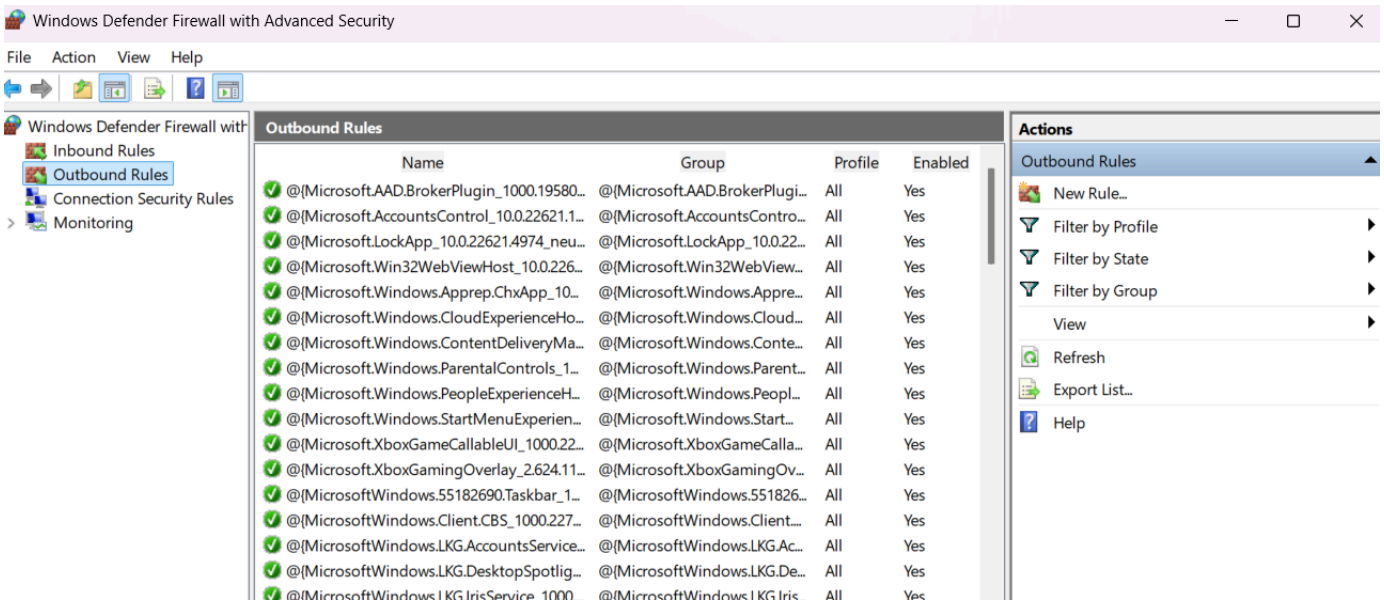
3. Click **“Advanced settings”** (opens **Windows Defender Firewall with Advanced Security**).

   In that MMC, open the **Inbound Rules** pane and then the **Outbound Rules** pane.

   Inbound Rules -



Outbound Rules -



Alternative quick open: Press Win + R, type wf.msc and press Enter, this opens the Advanced Security MMC directly.

Using **PowerShell (Administrator)**

1. Open PowerShell **as Administrator** (right-click > Run as administrator).
2. Run this to capture profile status and save to your Desktop:

Get-NetFirewallProfile | Format-Table Name, Enabled, DefaultInboundAction, DefaultOutboundAction -AutoSize | Out-File "$env:USERPROFILE\Desktop\fw_profile_status.txt"

```
PS C:\WINDOWS\system32> Get-NetFirewallProfile | Format-Table Name, Enabled, DefaultInboundAction, DefaultOutboundAction
 -AutoSize | Out-File "$env:USERPROFILE\Desktop\fw_profile_status.txt"
PS C:\WINDOWS\system32>
```

This creates fw_profile_status.txt on your Desktop with the baseline firewall profile states.

Step 2 List all existing firewall rules

Now we'll **export the current rule set** so you can show what existed before making any changes. This is useful for before/after comparison in your internship report.

In the Powershell run the following command:
Get-NetFirewallRule | Select-Object DisplayName, Direction, Action, Enabled, Profile | Export-Csv "$env:USERPROFILE\Desktop\fw_rules_baseline.csv" -NoTypeInformation

```
PS C:\WINDOWS\system32> Get-NetFirewallRule | Select-Object DisplayName, Direction, Action, Enabled, Profile | Export-Cs
v "$env:USERPROFILE\Desktop\fw_rules_baseline.csv" -NoTypeInformation
>>
```

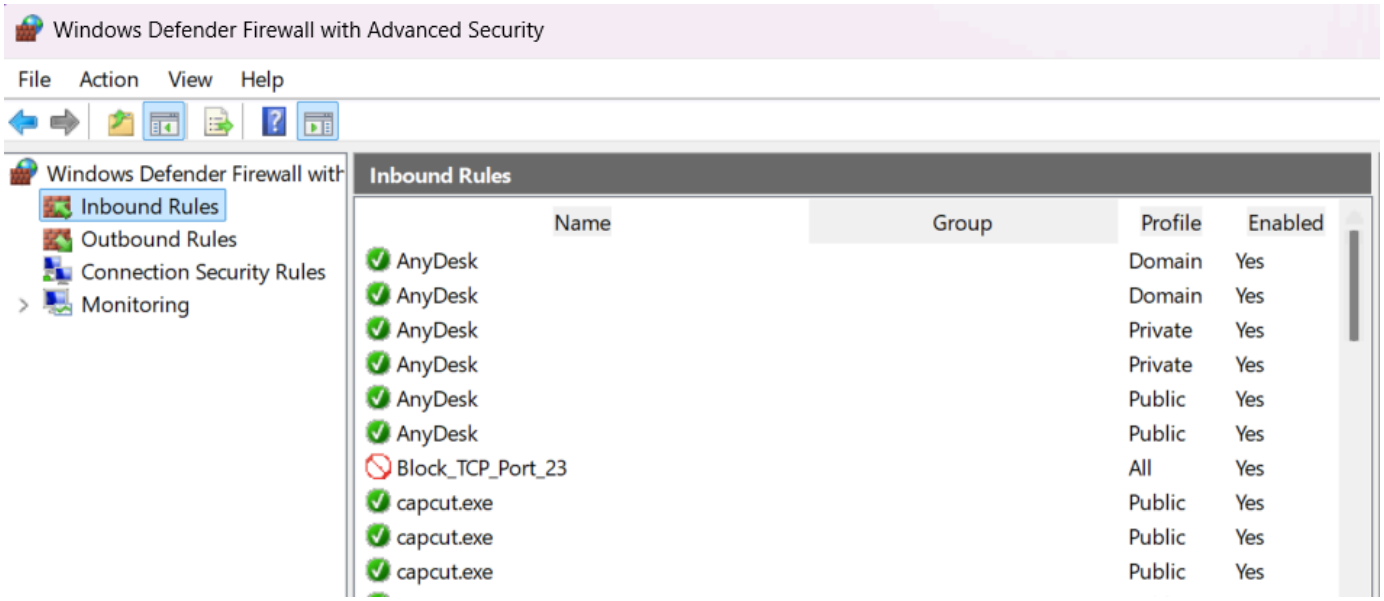
- This creates fw_rules_baseline.csv on your Desktop.
- The file will list each rule's name, direction (Inbound/Outbound), action (Allow/Block), whether it's enabled, and the profile it applies to.

Step 3 Create a custom firewall rule to block a specific port

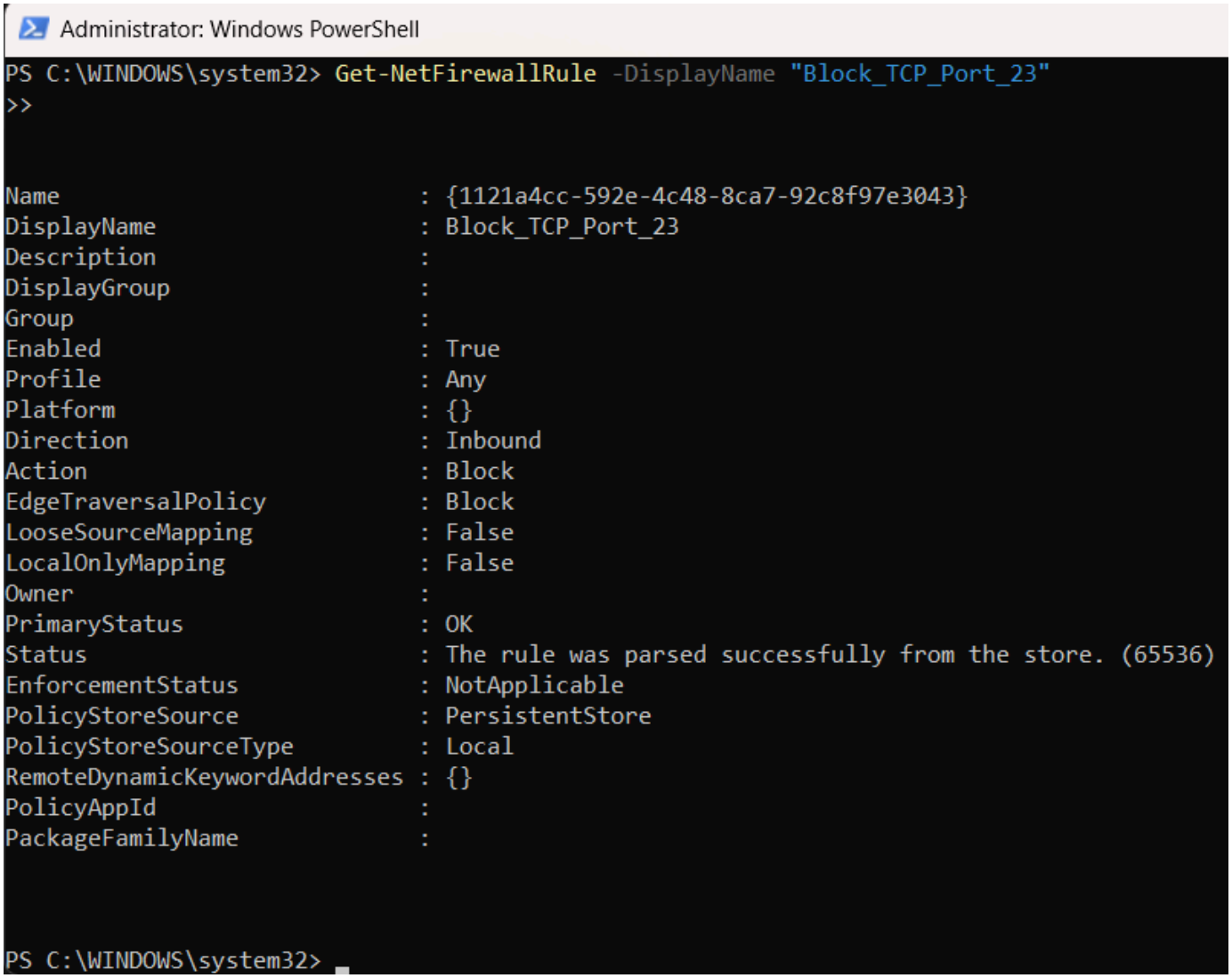
We'll block **TCP port 23** (commonly used for Telnet) in **Inbound traffic**.
This is safe for testing because Telnet is rarely used today, and it's a good example of targeted port blocking.

Open **PowerShell (run as Administrator)** and run the following command:
New-NetFirewallRule -DisplayName "Block_TCP_Port_23" -Direction Inbound -Protocol TCP -LocalPort 23 -Action Block

After running it, go to **Windows Defender Firewall with Advanced Security** > **Inbound Rules** and confirm we can see the new rule named **Block_TCP_Port_23**.



We can also verify via PowerShell using the following command:



Step 4 Test the firewall rule

We'll check if port **23** is actually blocked. Since Telnet is disabled by default on Windows, we'll use a simple method to confirm that the firewall is intercepting traffic.

Use PowerShell's Test-NetConnection by typing the following command:

Test-NetConnection -ComputerName 127.0.0.1 -Port 23

```
PS C:\WINDOWS\system32> Test-NetConnection -ComputerName 127.0.0.1 -Port 23
>>
WARNING: TCP connect to (127.0.0.1 : 23) failed


ComputerName            : 127.0.0.1
RemoteAddress           : 127.0.0.1
RemotePort              : 23
InterfaceAlias          : Loopback Pseudo-Interface 1
SourceAddress           : 127.0.0.1
PingSucceeded           : True
PingReplyDetails (RTT)  : 0 ms
TcpTestSucceeded        : False



PS C:\WINDOWS\system32>
```

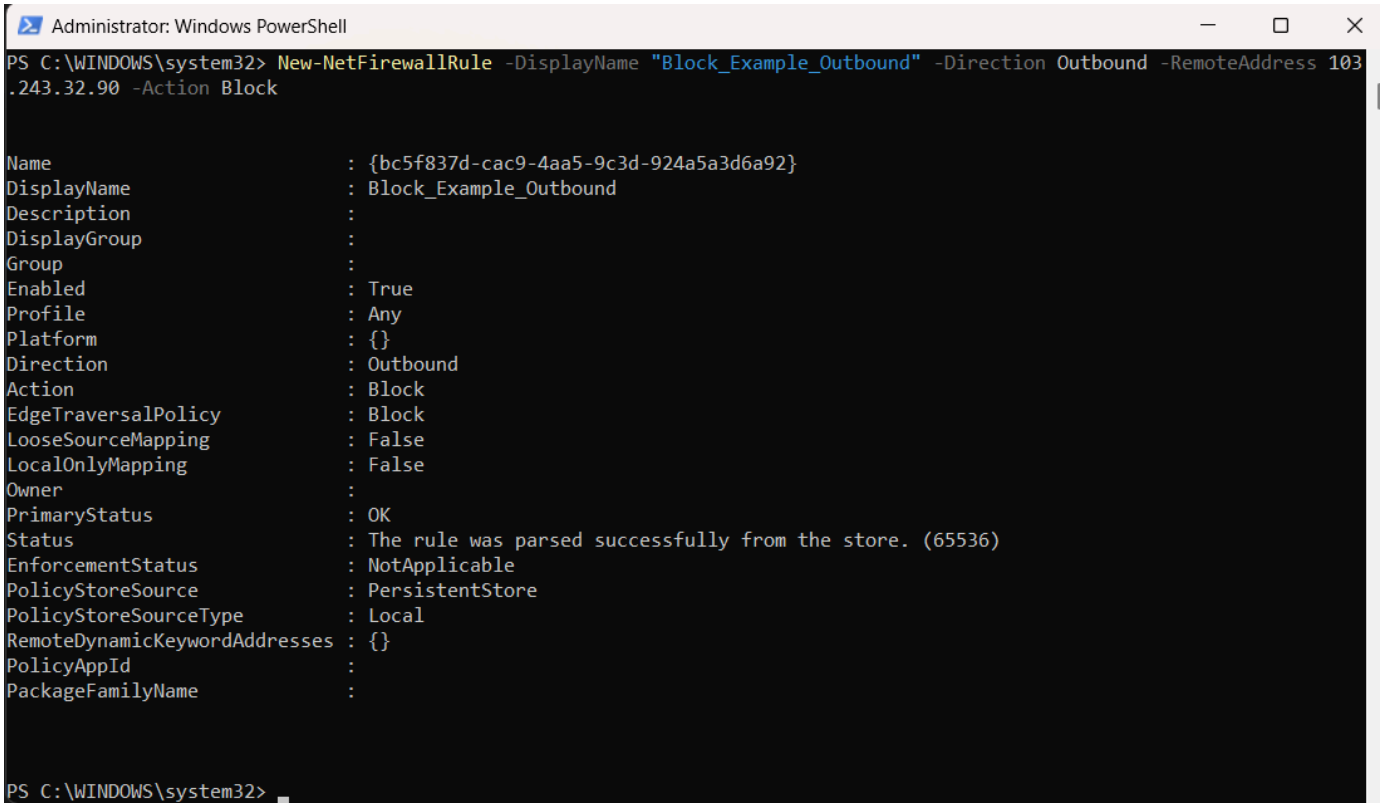
Step 5 Create and test an outbound firewall rule

In this, we'll Block outbound access to a specific website like flipkart.com. First, we have to know it's IP address so type the following command to figure out the target:

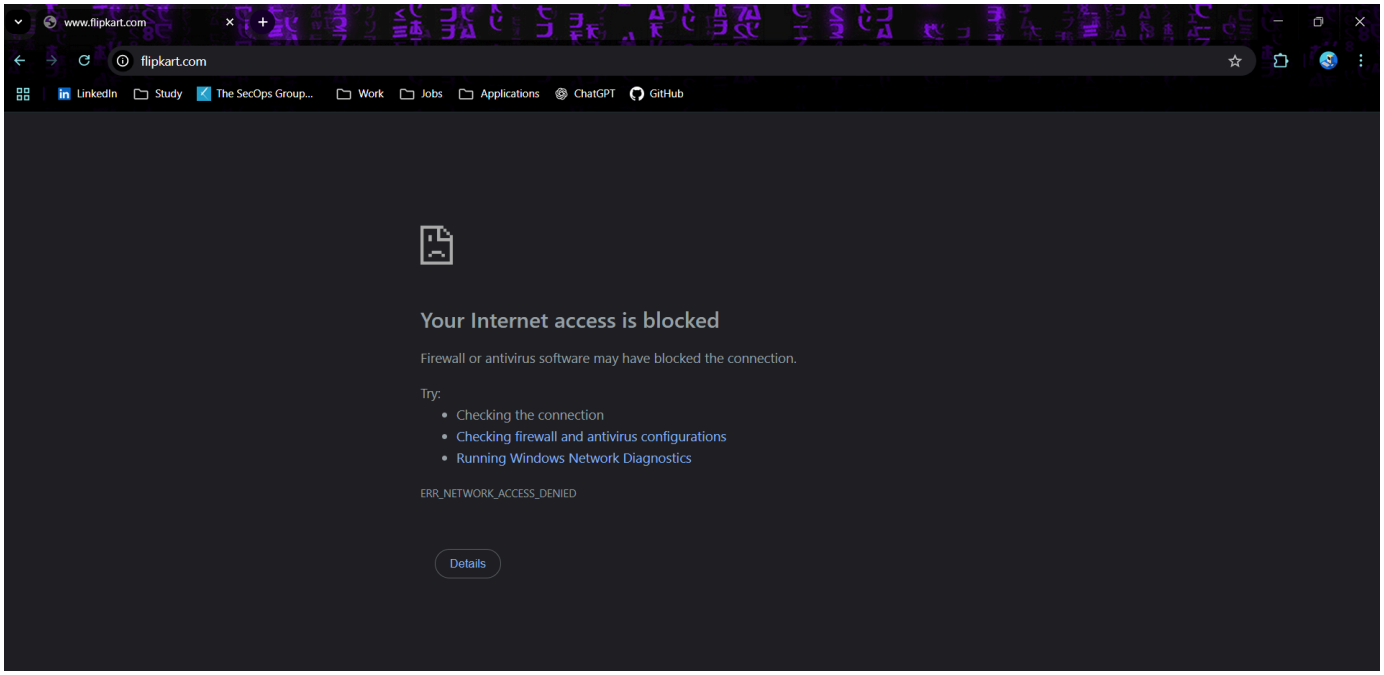
nslookup flipkart.com

```
PS C:\WINDOWS\system32> nslookup flipkart.com
Server:  reliance.reliance
Address:  192.168.29.1

Non-authoritative answer:
Name:    flipkart.com
Address:  103.243.32.90

PS C:\WINDOWS\system32>
```

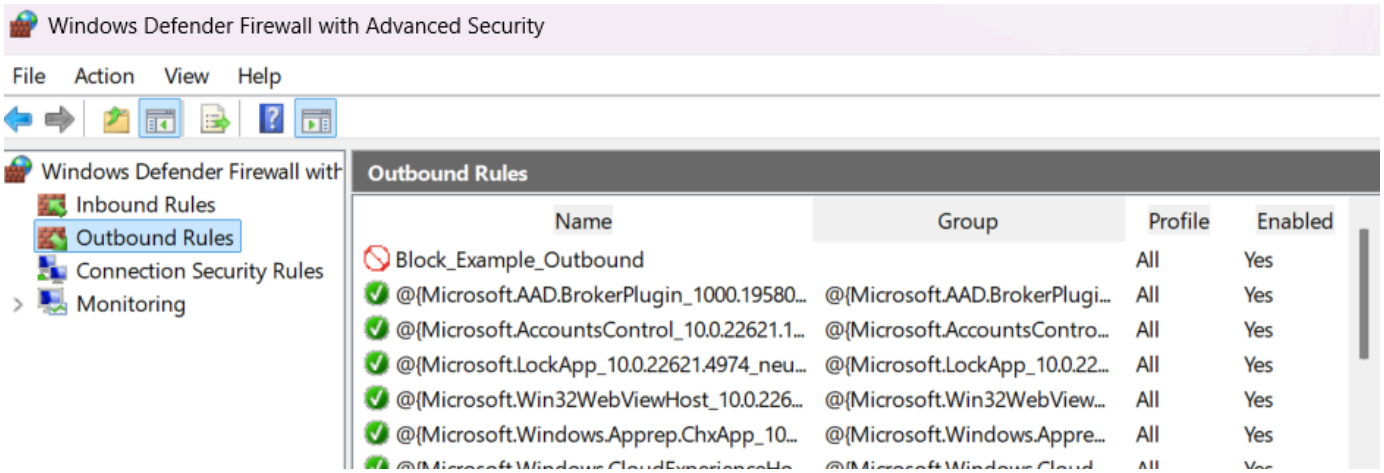
Now, create the outbound block rule using the following command:
New-NetFirewallRule -DisplayName "Block_Example_Outbound" -Direction Outbound -RemoteAddress 103.243.32.90 -Action Block



Let's test by trying to open the site in your browser, it should fail to load.



We can also verify **Windows Defender Firewall with Advanced Security** > **Outbound Rules** and confirm the new rule appears:



Finally to remove all this, type the following commands in PowerShell (Admin):

Remove-NetFirewallRule -DisplayName "Block_TCP_Port_23"

Remove-NetFirewallRule -DisplayName "Block_Flipkart_Outbound"

(Once they're removed, your Windows firewall will be back to the baseline state.)

Step 6 Enable and Configure UFW on Kali Linux

We'll now work on the **Linux** portion using **UFW (Uncomplicated Firewall)**, which is included in Kali but may not be enabled by default.

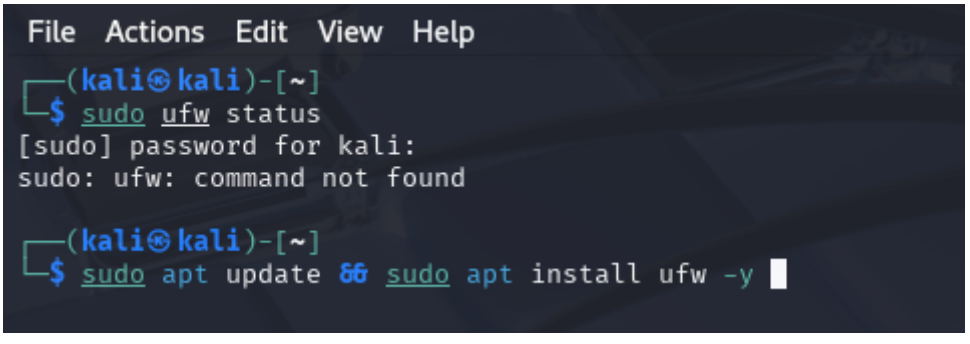
1. Check if UFW is installed

   In your Kali terminal:

   sudo ufw status

   If you get a message like command not found, install it:

   sudo apt update && sudo apt install ufw -y



2. Enable UFW by using the following command:
   sudo ufw enable

You might get a warning that it may disrupt SSH, if you're working directly on the Kali machine, it's fine to continue.

3. Check the current rules (baseline) by using the following command:
   sudo ufw status numbered

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw enable
sudo ufw status numbered

Firewall is active and enabled on system startup
Status: active
```

Step 7 Adding basic UFW allow/deny rules

1. Allow a specific service (HTTP on port 80)

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow 80/tcp

Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$
```

   This permits inbound HTTP traffic on TCP port 80.

2. Deny a specific port (Telnet on port 23)

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw deny 23/tcp

Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$
```

   This blocks inbound Telnet traffic.

3. Check and capture the rules

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw status numbered

Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 80/tcp                     ALLOW IN    Anywhere
[ 2] 23/tcp                     DENY IN     Anywhere
[ 3] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 4] 23/tcp (v6)                DENY IN     Anywhere (v6)


┌──(kali㉿kali)-[~]
└─$
```

Step 8 Test the UFW rules

In this step, we'll verify that: Port **80** is allowed and Port **23** is blocked.

```
┌──(kali㉿kali)-[~]
└─$ nc -zv 127.0.0.1 80
nc -zv 127.0.0.1 23

localhost [127.0.0.1] 80 (http) open
localhost [127.0.0.1] 23 (telnet) : Connection refused

┌──(kali㉿kali)-[~]
└─$
```

Step 9 Add advanced UFW configuration

1. Limit connections to prevent brute-force attacks

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw limit ssh

[sudo] password for kali:
Rule added
Rule added (v6)

┌──(kali㉿kali)-[~]
└─$
```

   This allows SSH but will block IPs with too many failed login attempts.

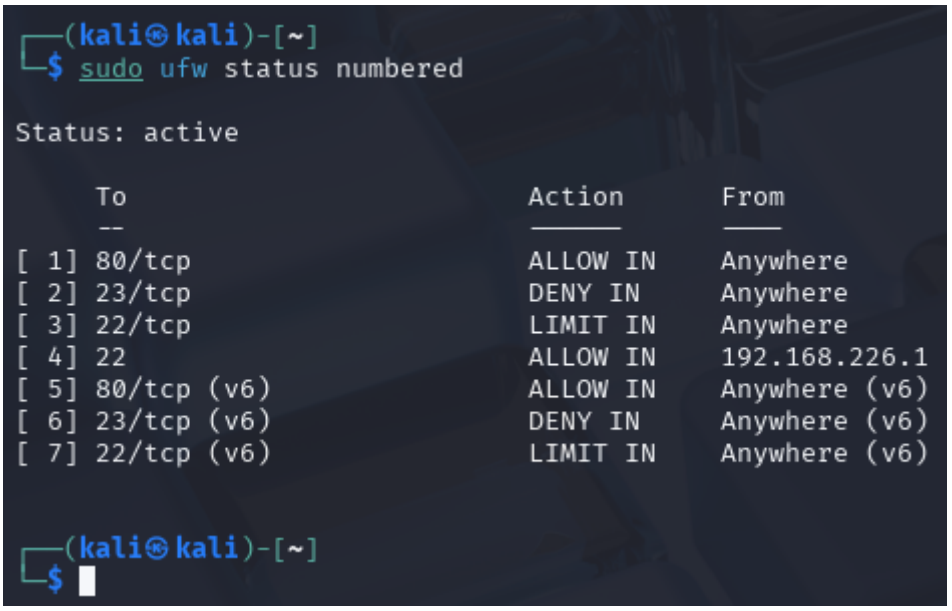
2. Allow a port for a specific IP only

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow from 192.168.226.1 to any port 22

Rule added

┌──(kali㉿kali)-[~]
└─$
```

3. Deny all incoming by default, allow all outgoing

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw default deny incoming
sudo ufw default allow outgoing

Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)

┌──(kali㉿kali)-[~]
└─$
```

4. Review the rules



To reset UFW so all the rules you added are removed and it’s back to default settings follow the given command -

1. Reset UFW to default:

   sudo ufw reset

   (This will disable UFW and delete all rules.)

2. Re-enable UFW with default policy:

   sudo ufw enable

   sudo ufw default allow incoming

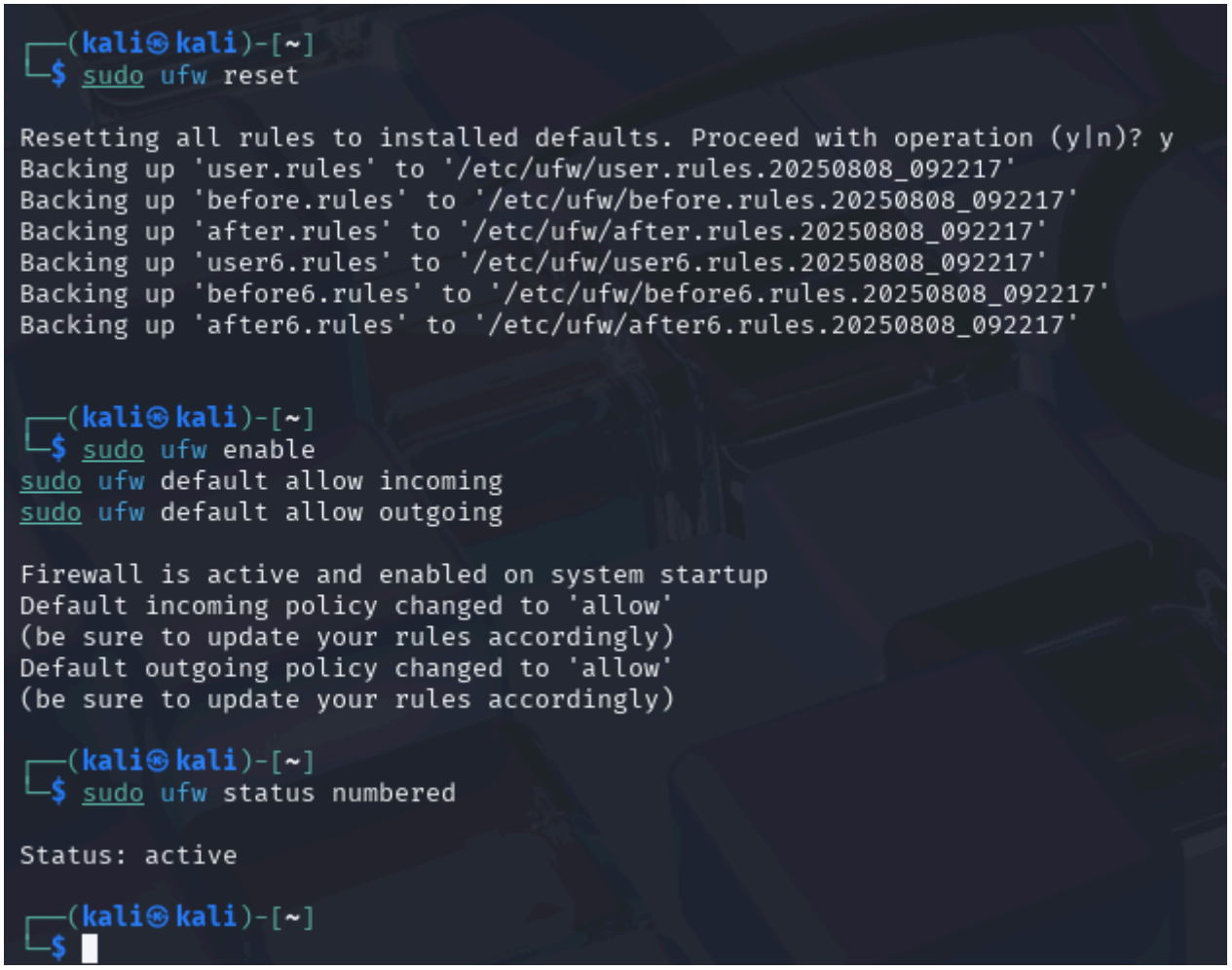
   sudo ufw default allow outgoing

   (This makes it fully open, like no blocking.)

3. Verify:

   sudo ufw status numbered

   (It should show **Status: active** and no custom rules.)



Outcome

By following the above steps:

- Baseline firewall configurations for both Windows and Linux were captured.
- Custom rules were created to block and allow traffic based on ports, IPs, and direction.
- Rules were tested using Test-NetConnection, nmap, and nc.
- Firewalls were reset to default settings after testing.

This process ensures understanding of firewall configuration, network traffic filtering, port control, UFW usage, and Windows Firewall management.