Tutorial 4 Lab Report
IPTables Firewall Configuration Experiment

Student Name: GuYi

October 26, 2025

1 Experiment Objectives

This experiment aims to learn the configuration and management of iptables firewall in Linux systems through practical operations, including:

- Understanding the basic concepts and working principles of iptables
- Mastering the creation, modification, and deletion of firewall rules
- Learning control methods for different types of network traffic
- Understanding the priority and matching mechanisms of firewall rules
- Practicing the implementation of network security protection strategies

2 Experiment Environment

- Operating System: Linux (Ubuntu/CentOS)
- Firewall Tool: iptables
- Testing Tools: ping, curl, nslookup, netcat, etc.
- Virtual Machine Environment: Safe experimental environment

3 Experiment Steps and Results

3.1 Preparation Phase - Environment Cleanup

Before starting the experiment, we need to clean up existing iptables rules to ensure we start from a clean state. This involves switching to root user, backing up current rules, cleaning all existing rules, and setting permissive policies for all chains.

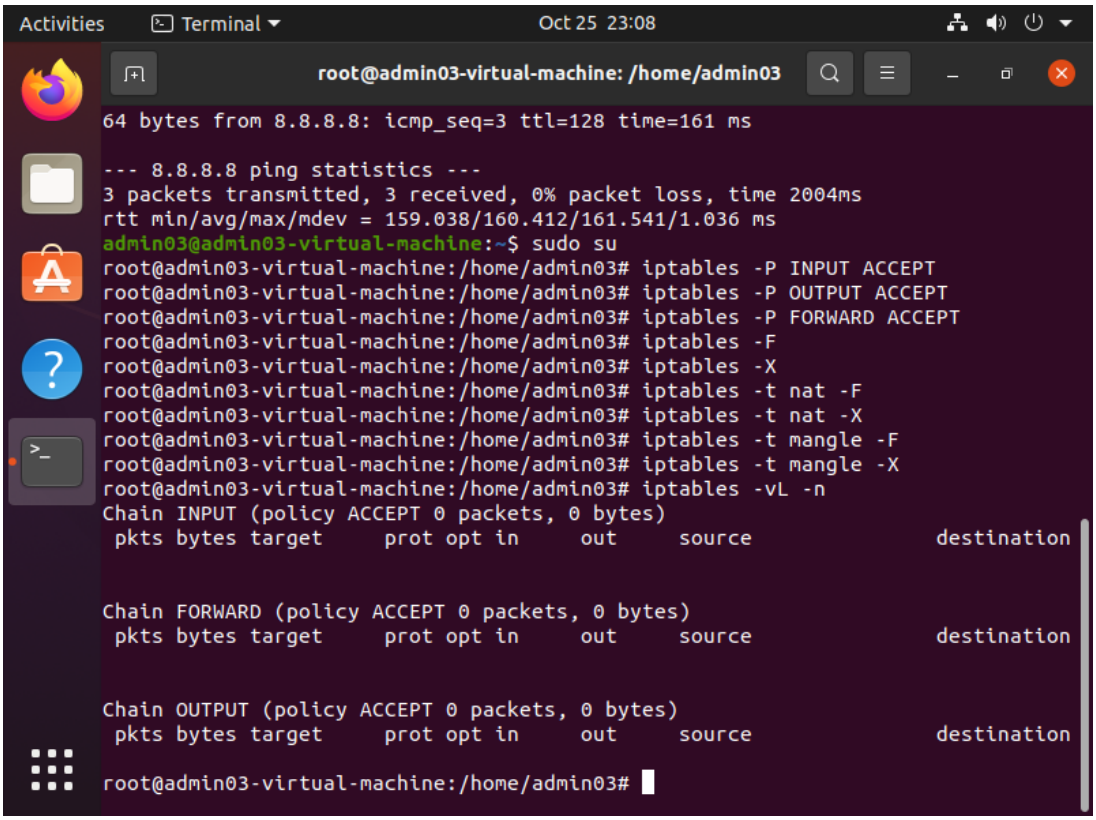

Figure 1: IPTables state after cleanup - showing all chains are empty with ACCEPT policy

From Figure 1, we can see that after cleanup, iptables shows that all three main chains (INPUT, FORWARD, OUTPUT) have no rules, and the default policies are all set to ACCEPT, providing a clean starting environment for subsequent experiments.

3.2 Rule 1: Setting Default Policies

Setting the default policy of the firewall is the first step in firewall configuration, usually adopting a "default deny" security strategy. We set the INPUT chain default policy to DROP (deny) and OUTPUT chain default policy to ACCEPT (allow).

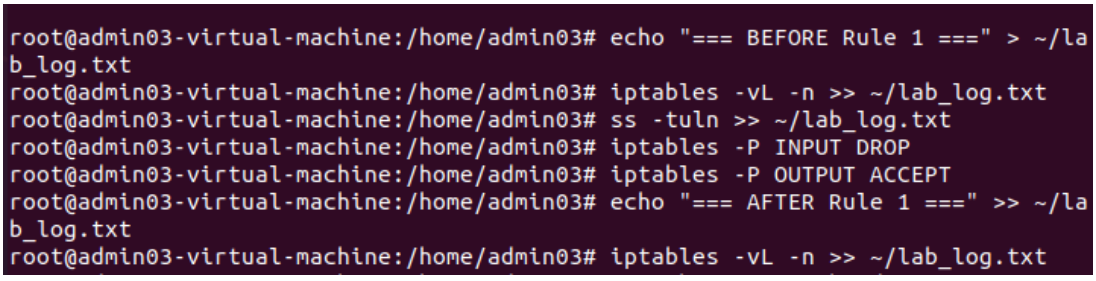

Figure 2: IPTables state after setting default policies - INPUT policy is DROP, OUTPUT policy is ACCEPT

Figure 2 shows the state after setting default policies. The INPUT chain policy becomes DROP, meaning all inbound traffic is denied by default; the OUTPUT chain re-

mains ACCEPT, allowing all outbound traffic. This is a common security configuration strategy.

3.3 Rule 2: Allow Loopback Communication

The loopback interface is used for internal machine communication and must remain open to ensure normal system operation. We configure rules to allow both inbound and outbound traffic on the loopback interface and test the communication.

```
root@admin03-virtual-machine:/home/admin03# iptables -A INPUT -i lo -j ACCEPT
root@admin03-virtual-machine:/home/admin03# iptables -A OUTPUT -o lo -j ACCEPT
root@admin03-virtual-machine:/home/admin03# ping -c 3 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.183 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.030 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.032 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.030/0.081/0.183/0.071 ms
```

Figure 3: Loopback interface rule configuration and test results

Figure 3 shows the configuration results of loopback interface rules. We can see that two new rules have been added to iptables, allowing inbound and outbound traffic on the loopback interface respectively. The ping test results show that local loopback communication works normally.

3.4 Rule 3: Allow ICMP Protocol

ICMP protocol is used for network diagnostics and error reporting, including the commonly used ping command. We configure rules to allow both inbound and outbound ICMP traffic and test ping to external networks.

```
root@admin03-virtual-machine:/home/admin03# iptables -A INPUT -p icmp -j ACCEPT
root@admin03-virtual-machine:/home/admin03# iptables -A OUTPUT -p icmp -j ACCEP
T
root@admin03-virtual-machine:/home/admin03# echo "=== AFTER Rule 3 ===" >> ~/la
b_log.txt
root@admin03-virtual-machine:/home/admin03# iptables -vL -n >> ~/lab_log.txt
root@admin03-virtual-machine:/home/admin03# ping -c 3 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=161 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=160 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=161 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
```

Figure 4: ICMP protocol rule configuration and ping test results

Figure 4 shows the state after ICMP rule configuration. Rules allowing ICMP protocol have been added to iptables, and the ping test to 8.8.8.8 was successful, proving that ICMP traffic can pass through the firewall normally.

3.5 Rule 4: Allow Outbound Web Access

Configure outbound access for HTTP and HTTPS protocols, which is a basic requirement for modern network applications. We allow outbound connections on ports 80 and 443, and configure corresponding inbound rules for established connections using state tracking.

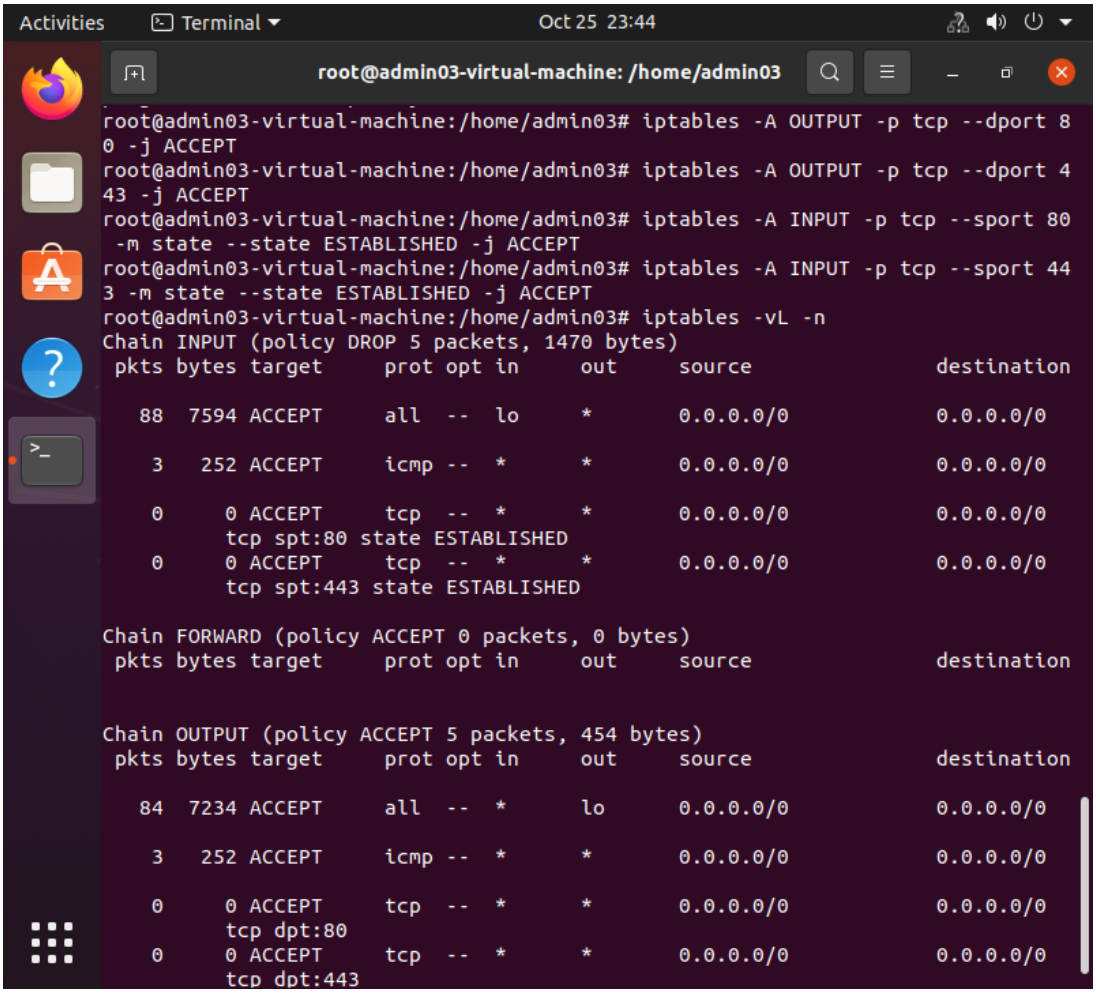

Figure 5: Web access rule configuration results

Figure 5 shows the configuration of web access rules. We can see four new rules have been added, handling HTTP and HTTPS outbound requests and their corresponding inbound responses. The state tracking mechanism (state ESTABLISHED) is used to allow response packets from established connections.

3.6 Rule 5: Enable DNS Resolution

DNS resolution is the foundation of network access, requiring configuration of access rules for UDP port 53. We allow DNS queries and test both DNS resolution and web access functionality.

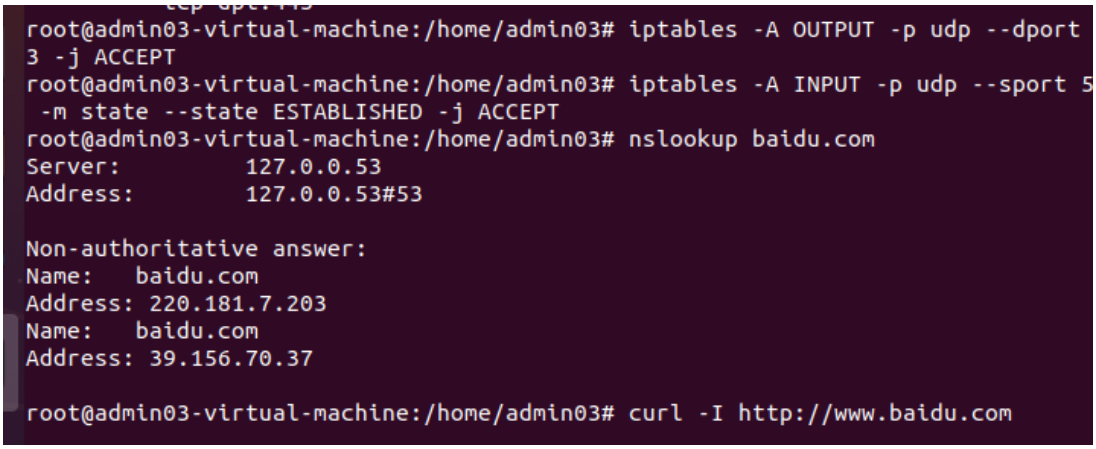

Figure 6: DNS rule configuration and test results

Figure 6 shows the test results after DNS rule configuration. The nslookup command successfully resolved the baidu.com domain name, and the curl command was also able to access the website normally, indicating that both DNS resolution and web access functions work properly.

3.7 Rule 6: Block Specific Website Access

Demonstrate how to block access to specific websites, which is a basic implementation of content filtering. We find the IP address of the target website, block access to that specific IP, and test the blocking effect.

```
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   baidu.com
Address: 39.156.70.37
Name:   baidu.com
Address: 220.181.7.203

root@admin03-virtual-machine:/home/admin03# iptables -I OUTPUT -d 39.156.70.37
-j DROP
root@admin03-virtual-machine:/home/admin03# iptables -I OUTPUT -d 220.181.7.203
 -j DROP
root@admin03-virtual-machine:/home/admin03# curl -I http://baidu.com

Caution: You are using the Snap version of curl.
Due to Snap's sandbox nature, this version has some limitations.
For example, it may not be able to access hidden folders in your home directory
or other restricted areas of the os.
This means you may encounter errors when using snap curl to download files.
For those case, you might want to use the native curl package.
For details, see: https://github.com/boukendesho/curl-snap/issues/1

To stop seeing this message, run the following command:
curl.snap-acked

^C
root@admin03-virtual-machine:/home/admin03# curl.snap-acked
You will no longer see the Snap warning message.
root@admin03-virtual-machine:/home/admin03# curl -I http://baidu.com

curl: (28) Failed to connect to baidu.com port 80 after 131270 ms: Could not co
nnect to server
root@admin03-virtual-machine:/home/admin03#
```

Figure 7: Website blocking rule configuration and test results

Figure 7 shows the implementation of website blocking functionality. We obtained the IP address of facebook.com through nslookup, then used iptables rules to block access to that IP. Test results show that access to facebook.com is blocked, while access to baidu.com remains normal.

3.8 Rule 7-8: Complete Rule Configuration

Add support for other commonly used protocols, such as SSH. We allow outbound SSH connections and configure corresponding inbound rules for established connections, then view the complete rule configuration.

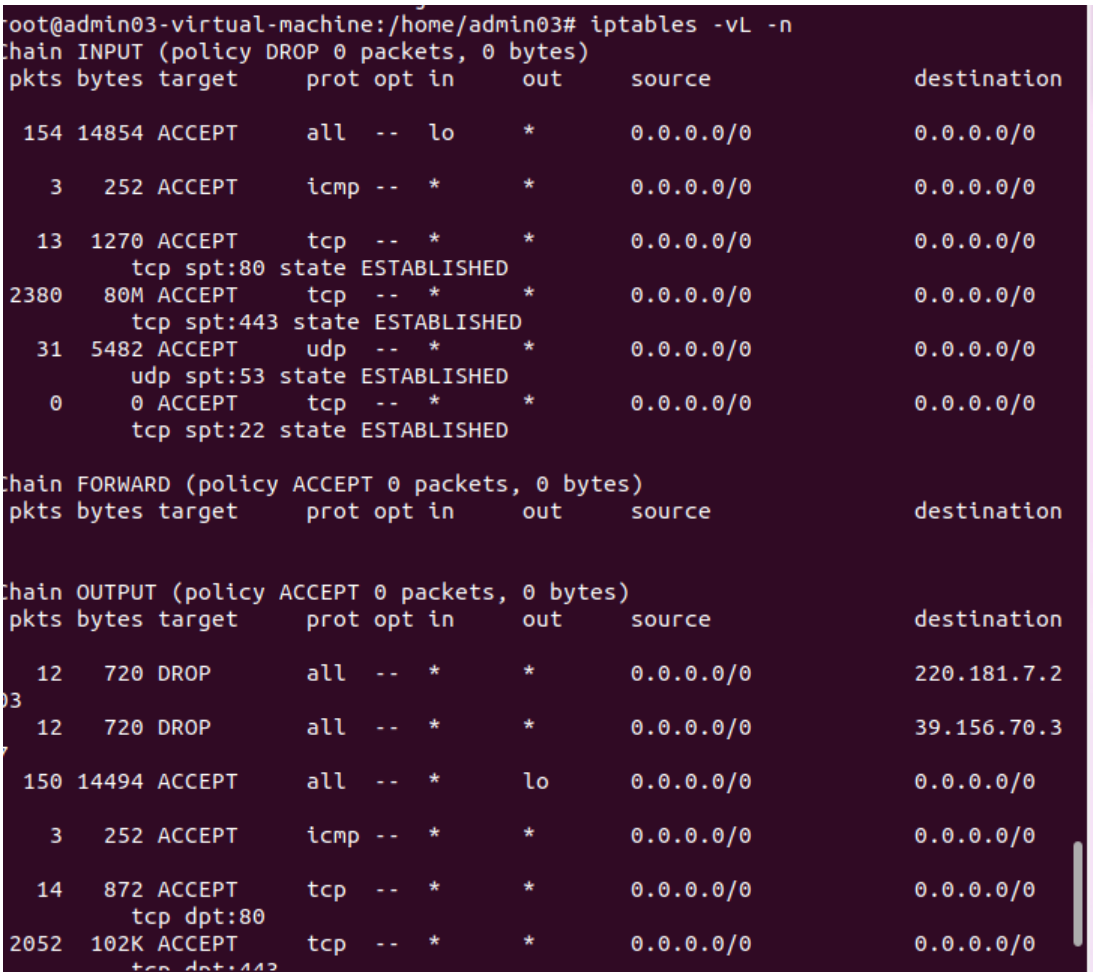

Figure 8: Complete firewall rule configuration

Figure 8 shows the complete firewall rule configuration. We can see that the INPUT chain contains rules for loopback, ICMP, web responses, DNS responses, and SSH responses; the OUTPUT chain contains rules for loopback, ICMP, web requests, DNS requests, SSH requests, and website blocking.

3.9 Rule 9: Inbound Web Service Configuration

Configure web server to allow external access to the local web service. We install Apache web server, allow inbound HTTP access, start the Apache service, and test the local web service.

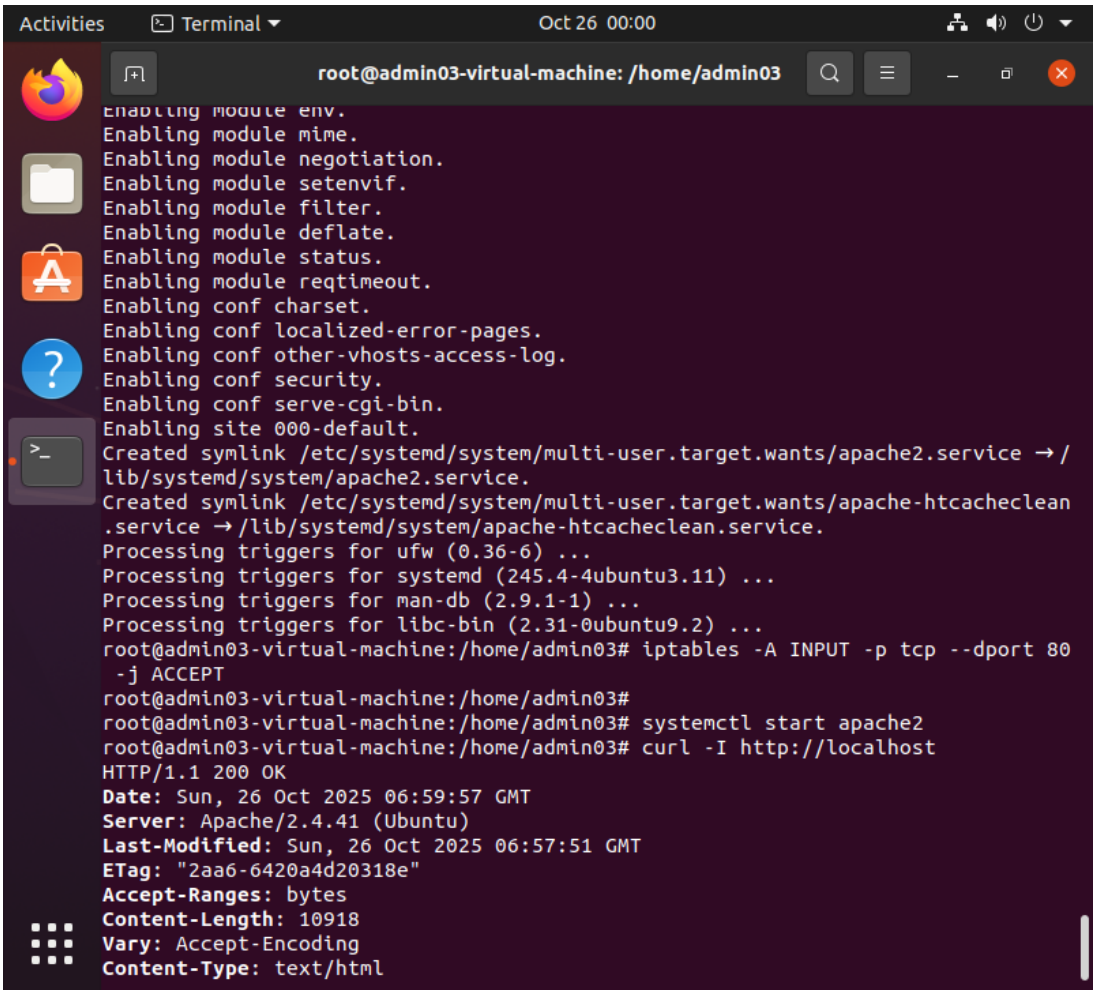

Figure 9: Inbound web service configuration and testing

Figure 9 shows the configuration process of inbound web service. The Apache server was successfully installed and started, iptables rules allow inbound HTTP connections, and the curl test shows that the local web service responds normally.

3.10 Rule 10: Rule Priority Testing

Demonstrate the priority mechanism of iptables rules, where the first matching rule determines packet processing. We add conflicting rules to test priority, view the rule order, and test port access.

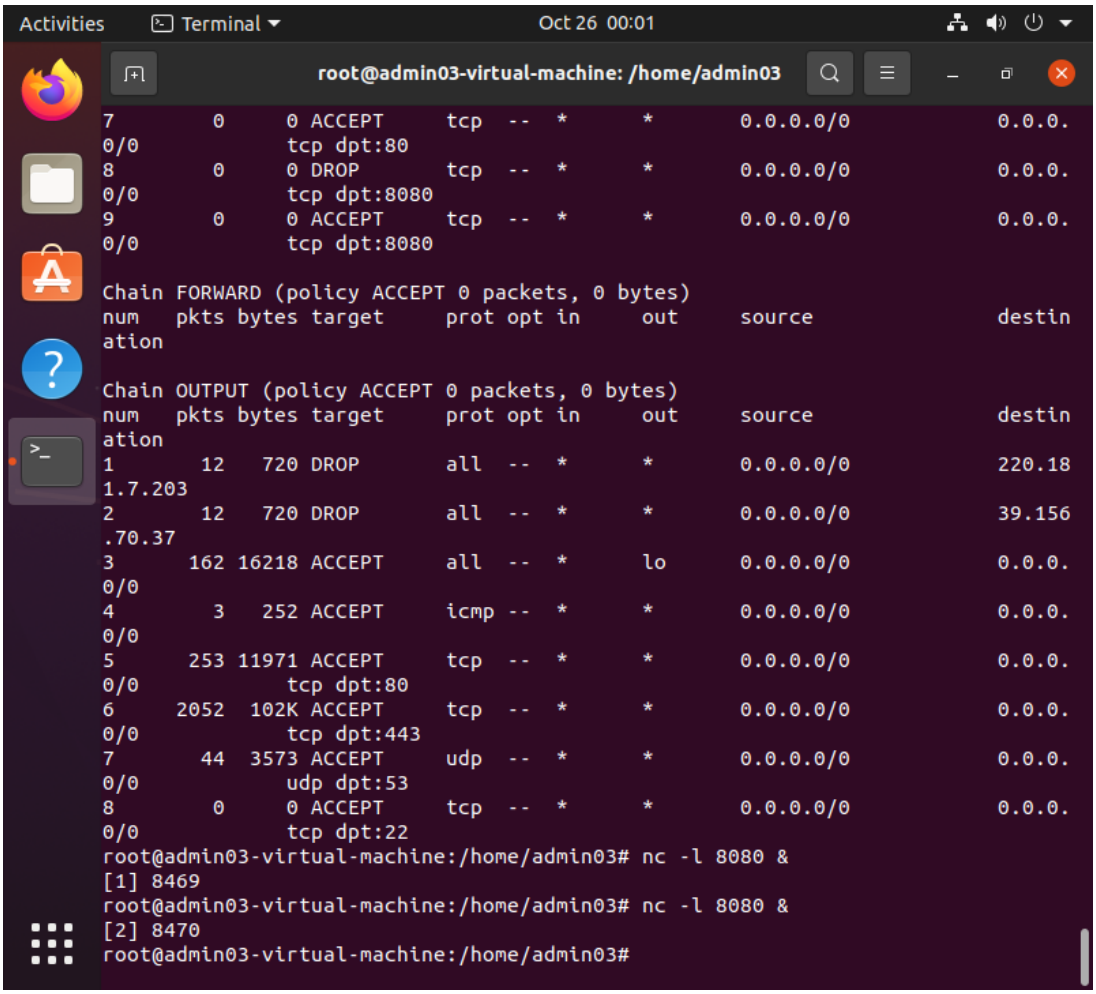

Figure 10: Rule priority test results

Figure 10 demonstrates the priority mechanism of iptables rules. Two conflicting rules were added: first DROP then ACCEPT for the same port. Since iptables matches rules in order, the first DROP rule takes effect, and the subsequent ACCEPT rule is not executed.

3.11 Final Configuration Check

View the final complete firewall configuration state. We examine the final complete rules with line numbers and view rule statistics.

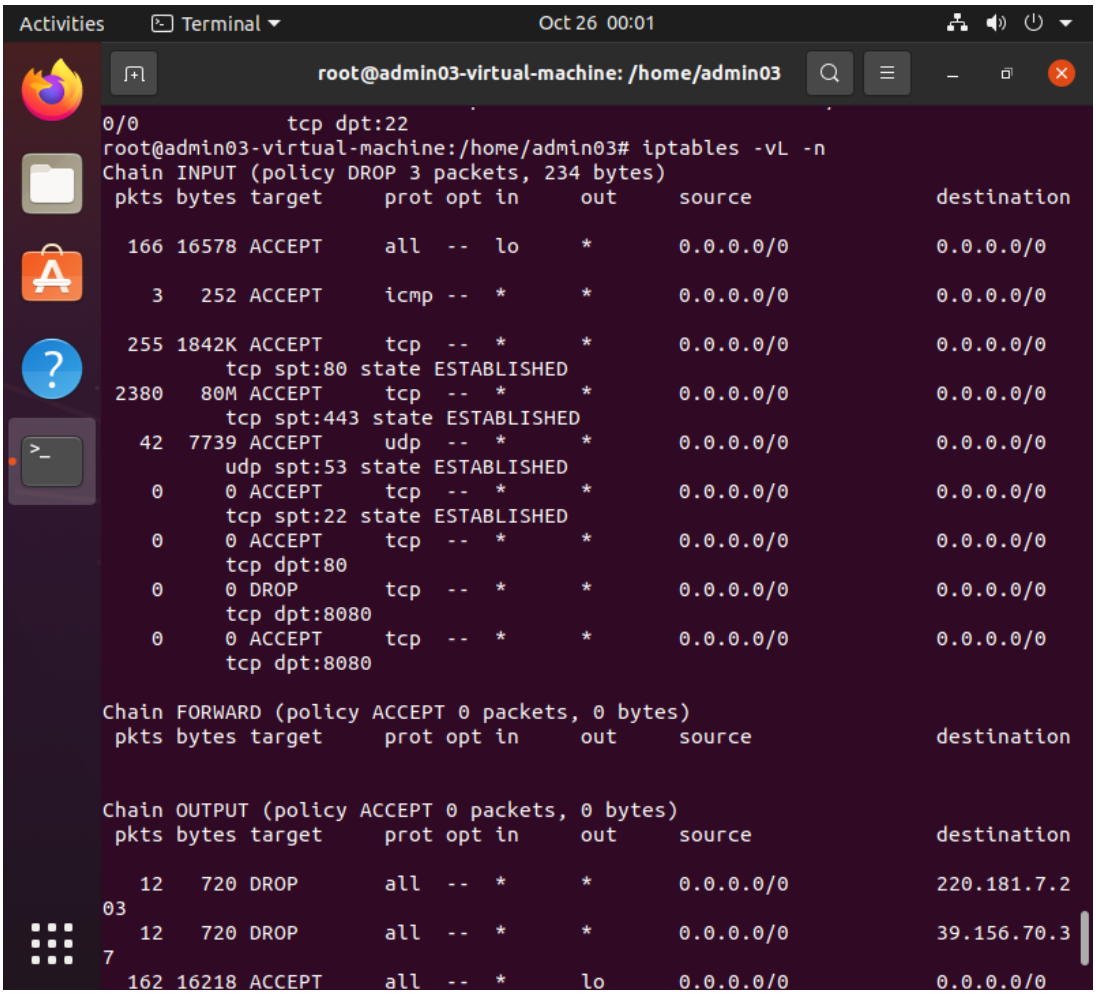

Figure 11: Final complete firewall configuration

Figure 11 shows the final firewall configuration after completing the experiment. We can see the complete rule set, including support for various protocols, website blocking, service access control, and other functions. Rule statistics show the match count and traffic statistics for each rule.

4 Experiment Summary and Analysis

4.1 Experiment Achievements

Through this experiment, the following tasks were successfully completed:

1. **Basic Configuration**: Mastered basic iptables operations, including rule cleanup, policy setting, etc.

2. **Protocol Control**: Learned to configure access rules for different network protocols (ICMP, HTTP, HTTPS, DNS, SSH)

3. **Access Control**: Implemented website blocking functionality, understanding IP address-based access control

4. **Service Configuration**: Configured inbound access rules for web servers

5. **Rule Priority**: Understood the matching order and priority mechanism of iptables rules

4.2 Key Technical Points

- **Default Policy**: Adopting "default deny" policy to improve security
- **State Tracking**: Using connection tracking mechanism to manage connection states
- **Rule Order**: IPTables matches rules in order, the first matching rule determines processing
- **Port Control**: Controlling different types of network traffic through source and destination ports
- **Interface Control**: Using network interface parameters to control traffic on specific interfaces

4.3 Security Considerations

- Loopback interface must remain open, otherwise internal system communication will be affected
- SSH rule configuration requires caution to avoid locking out remote access
- Rule order is important, more specific rules should be placed first
- Regularly backup firewall rules for fault recovery
- Thorough testing should be conducted before applying rules in production environments

4.4 Experiment Insights

This experiment gave me a deep understanding of the working principles and configuration methods of Linux firewalls. IPTables, as an important security component of Linux systems, provides powerful protection for network security through its flexibility and powerful functionality. Through practical operations, I mastered the design concepts and implementation methods of firewall rules, which is of great significance for system administration and network security work.

The main challenges encountered during the experiment were understanding the rule matching logic and state tracking mechanisms. Through repeated testing and log observation, I finally mastered these key concepts.

5 References

- Linux iptables Official Documentation
- "Linux Firewall Configuration Guide"

- Tutorial 4 Lab Guide Document
- Network Security Related Technical Materials