Nftables lab

**NFTables Command Labs - Basic to Intermediate**

**Prerequisites**

* **Linux system with nftables installed**
* **Root or sudo privileges**
* **Basic understanding of network concepts**

**Lab Environment Setup**

**# Check if nftables is installed**

**nft --version**

**# Install nftables (if not installed)**

**# Ubuntu/Debian:**

**sudo apt install nftables**

**# CentOS/RHEL:**

**sudo yum install nftables**

**# or**

**sudo dnf install nftables**

**# Start and enable nftables service**

**sudo systemctl start nftables**

**sudo systemctl enable nftables**

**Lab 1: Basic NFTables Concepts**

**1.1 Understanding NFTables Structure**

**# View current ruleset**

**sudo nft list ruleset**

**# View all tables**

**sudo nft list tables**

**# View table families**

**sudo nft list tables inet**

**sudo nft list tables ip**

**sudo nft list tables ip6**

**1.2 Creating Your First Table**

**# Create a basic inet table (handles both IPv4 and IPv6)**

**sudo nft add table inet filter**

**# Create IPv4 specific table**

**sudo nft add table ip filter\_ipv4**

**# Create IPv6 specific table**

**sudo nft add table ip6 filter\_ipv6**

**# List tables to verify**

**sudo nft list tables**

**1.3 Basic Chain Operations**

**# Create a basic input chain**

**sudo nft add chain inet filter input { type filter hook input priority 0 \; }**

**# Create output chain**

**sudo nft add chain inet filter output { type filter hook output priority 0 \; }**

**# Create forward chain**

**sudo nft add chain inet filter forward { type filter hook forward priority 0 \; }**

**# List chains in a table**

**sudo nft list table inet filter**

**Lab 2: Basic Filtering Rules**

**2.1 Allow/Drop Rules**

**# Allow loopback traffic**

**sudo nft add rule inet filter input iif lo accept**

**# Allow established and related connections**

**sudo nft add rule inet filter input ct state established,related accept**

**# Drop invalid packets**

**sudo nft add rule inet filter input ct state invalid drop**

**# Allow SSH (port 22)**

**sudo nft add rule inet filter input tcp dport 22 accept**

**# Allow HTTP and HTTPS**

**sudo nft add rule inet filter input tcp dport { 80, 443 } accept**

**# Drop all other input (default policy)**

**sudo nft add rule inet filter input drop**

**2.2 Source/Destination IP Filtering**

**# Allow from specific IP**

**sudo nft add rule inet filter input ip saddr 192.168.1.100 accept**

**# Allow from IP range**

**sudo nft add rule inet filter input ip saddr 192.168.1.0/24 accept**

**# Block specific IP**

**sudo nft add rule inet filter input ip saddr 10.0.0.50 drop**

**# Allow to specific destination**

**sudo nft add rule inet filter output ip daddr 8.8.8.8 accept**

**2.3 Protocol Filtering**

**# Allow ICMP ping**

**sudo nft add rule inet filter input icmp type echo-request accept**

**# Allow ICMPv6**

**sudo nft add rule inet filter input icmpv6 type { echo-request, nd-neighbor-solicit, nd-neighbor-advert } accept**

**# Allow UDP DNS queries**

**sudo nft add rule inet filter input udp dport 53 accept**

**# Allow specific protocols**

**sudo nft add rule inet filter input ip protocol { tcp, udp } accept**

**Lab 3: Intermediate Rule Management**

**3.1 Rule Positioning and Modification**

**# Insert rule at specific position (position 1)**

**sudo nft insert rule inet filter input position 1 tcp dport 22 accept**

**# Add rule after specific rule (get handle first)**

**sudo nft --handle list table inet filter**

**sudo nft add rule inet filter input position 5 tcp dport 80 accept**

**# Replace a rule (use handle number)**

**sudo nft replace rule inet filter input handle 10 tcp dport 8080 accept**

**# Delete specific rule by handle**

**sudo nft delete rule inet filter input handle 10**

**3.2 Working with Sets**

**# Create a set of IP addresses**

**sudo nft add set inet filter blacklist { type ipv4\_addr \; }**

**# Add elements to set**

**sudo nft add element inet filter blacklist { 192.168.1.50, 192.168.1.51 }**

**# Use set in rules**

**sudo nft add rule inet filter input ip saddr @blacklist drop**

**# Create set with timeout**

**sudo nft add set inet filter temp\_block { type ipv4\_addr \; timeout 1h \; }**

**# Create set from file**

**echo "192.168.1.100" > /tmp/allowed\_ips.txt**

**echo "192.168.1.101" >> /tmp/allowed\_ips.txt**

**sudo nft add set inet filter allowed\_ips { type ipv4\_addr \; }**

**sudo nft add element inet filter allowed\_ips { $(cat /tmp/allowed\_ips.txt | tr '\n' ',' | sed 's/,$//') }**

**3.3 Advanced Port Handling**

**# Port ranges**

**sudo nft add rule inet filter input tcp dport 1024-65535 accept**

**# Multiple ports and ranges**

**sudo nft add rule inet filter input tcp dport { 22, 80, 443, 8000-8080 } accept**

**# Source and destination ports**

**sudo nft add rule inet filter input tcp sport 80 dport 1024-65535 accept**

**# Dynamic port ranges**

**sudo nft add rule inet filter input tcp dport != 22 drop**

**Lab 4: Connection Tracking and Stateful Filtering**

**4.1 Connection States**

**# Track connection states**

**sudo nft add rule inet filter input ct state new,established,related accept**

**# Allow new connections only from specific networks**

**sudo nft add rule inet filter input ip saddr 192.168.1.0/24 ct state new accept**

**# Drop invalid connections**

**sudo nft add rule inet filter input ct state invalid counter drop**

**# Connection tracking with logging**

**sudo nft add rule inet filter input ct state new log prefix "NEW\_CONN: " accept**

**4.2 Connection Limiting**

**# Limit new connections per minute**

**sudo nft add rule inet filter input ct state new limit rate 10/minute accept**

**# Limit connections per source IP**

**sudo nft add rule inet filter input ct state new limit rate over 5/minute drop**

**# Burst limiting**

**sudo nft add rule inet filter input ct state new limit rate 10/minute burst 5 packets accept**

**4.3 Advanced Connection Tracking**

**# Track connections by mark**

**sudo nft add rule inet filter input ct mark 1 accept**

**# Set connection mark**

**sudo nft add rule inet filter input tcp dport 80 ct mark set 1**

**# Track helper modules (FTP example)**

**sudo nft add rule inet filter input tcp dport 21 ct helper set "ftp"**

**Lab 5: NAT and Address Translation**

**5.1 Basic NAT Setup**

**# Create NAT table**

**sudo nft add table ip nat**

**# Create prerouting chain for DNAT**

**sudo nft add chain ip nat prerouting { type nat hook prerouting priority -100 \; }**

**# Create postrouting chain for SNAT**

**sudo nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }**

**# Basic masquerading (SNAT)**

**sudo nft add rule ip nat postrouting oif eth0 masquerade**

**# Static SNAT**

**sudo nft add rule ip nat postrouting oif eth0 snat to 203.0.113.10**

**5.2 Port Forwarding (DNAT)**

**# Forward external port 8080 to internal server**

**sudo nft add rule ip nat prerouting iif eth0 tcp dport 8080 dnat to 192.168.1.100:80**

**# Forward SSH to internal server**

**sudo nft add rule ip nat prerouting iif eth0 tcp dport 2222 dnat to 192.168.1.100:22**

**# Port range forwarding**

**sudo nft add rule ip nat prerouting iif eth0 tcp dport 8000-8080 dnat to 192.168.1.100:8000-8080**

**5.3 Load Balancing NAT**

**# Round-robin load balancing**

**sudo nft add rule ip nat prerouting iif eth0 tcp dport 80 dnat to numgen random mod 3 map { 0 : 192.168.1.100, 1 : 192.168.1.101, 2 : 192.168.1.102 }**

**# Weighted load balancing**

**sudo nft add rule ip nat prerouting iif eth0 tcp dport 80 dnat to numgen random mod 10 map { 0-6 : 192.168.1.100, 7-9 : 192.168.1.101 }**

**Lab 6: Traffic Shaping and QoS**

**6.1 Traffic Classification**

**# Mark packets for QoS**

**sudo nft add table inet mangle**

**sudo nft add chain inet mangle prerouting { type filter hook prerouting priority -150 \; }**

**# Mark HTTP traffic**

**sudo nft add rule inet mangle prerouting tcp dport 80 meta mark set 1**

**# Mark by DSCP**

**sudo nft add rule inet mangle prerouting ip dscp 46 meta mark set 2**

**# Mark by packet size**

**sudo nft add rule inet mangle prerouting meta length > 1000 meta mark set 3**

**6.2 Rate Limiting**

**# Limit bandwidth per IP**

**sudo nft add rule inet filter input ip saddr 192.168.1.0/24 limit rate 1 mbytes/second accept**

**# Limit by connection count**

**sudo nft add rule inet filter input tcp dport 80 meter http\_conn { ip saddr limit rate over 10/minute } drop**

**# Burst control**

**sudo nft add rule inet filter input limit rate 100/second burst 50 packets accept**

**Lab 7: Logging and Monitoring**

**7.1 Basic Logging**

**# Log dropped packets**

**sudo nft add rule inet filter input counter log prefix "DROPPED: " drop**

**# Log with specific log level**

**sudo nft add rule inet filter input tcp dport 22 log level warn prefix "SSH\_ACCESS: " accept**

**# Log with rate limiting**

**sudo nft add rule inet filter input log prefix "INPUT: " limit rate 1/second accept**

**7.2 Advanced Monitoring**

**# Create named counters**

**sudo nft add counter inet filter http\_requests**

**sudo nft add counter inet filter ssh\_attempts**

**# Use counters in rules**

**sudo nft add rule inet filter input tcp dport 80 counter name http\_requests accept**

**sudo nft add rule inet filter input tcp dport 22 counter name ssh\_attempts accept**

**# View counter statistics**

**sudo nft list counters**

**Lab 8: Security Hardening**

**8.1 Anti-DDoS Measures**

**# SYN flood protection**

**sudo nft add rule inet filter input tcp flags syn limit rate 25/second accept**

**# Connection limit per source**

**sudo nft add rule inet filter input ct state new meter conn\_limit { ip saddr limit rate over 10/minute } drop**

**# Block ping floods**

**sudo nft add rule inet filter input icmp type echo-request limit rate 1/second accept**

**8.2 Geo-blocking and Advanced Filtering**

**# Create countries set (example)**

**sudo nft add set inet filter blocked\_countries { type ipv4\_addr \; flags interval \; }**

**# Block specific network ranges**

**sudo nft add element inet filter blocked\_countries { 223.0.0.0/8, 221.0.0.0/8 }**

**# Use in rules**

**sudo nft add rule inet filter input ip saddr @blocked\_countries drop**

**# Time-based rules**

**sudo nft add rule inet filter input tcp dport 22 meta hour "09:00"-"17:00" accept**

**Lab 9: Scripting and Automation**

**9.1 NFTables Scripts**

**# Create a script file**

**cat > /tmp/nft-rules.nft << 'EOF'**

**#!/usr/sbin/nft -f**

**# Clear existing rules**

**flush ruleset**

**# Create main table**

**table inet filter {**

**chain input {**

**type filter hook input priority 0; policy drop;**

**# Allow loopback**

**iif lo accept**

**# Allow established connections**

**ct state established,related accept**

**# Allow SSH**

**tcp dport 22 accept**

**# Allow HTTP/HTTPS**

**tcp dport { 80, 443 } accept**

**# Log and drop everything else**

**log prefix "DROPPED: " drop**

**}**

**chain output {**

**type filter hook output priority 0; policy accept;**

**}**

**}**

**EOF**

**# Apply the script**

**sudo nft -f /tmp/nft-rules.nft**

**9.2 Dynamic Rule Management**

**# Function to block IP temporarily**

**block\_ip() {**

**local ip=$1**

**local duration=${2:-3600} # Default 1 hour**

**# Add to temporary set**

**sudo nft add element inet filter temp\_block { $ip timeout ${duration}s }**

**echo "Blocked $ip for $duration seconds"**

**}**

**# Function to unblock IP**

**unblock\_ip() {**

**local ip=$1**

**sudo nft delete element inet filter temp\_block { $ip }**

**echo "Unblocked $ip"**

**}**

**# Usage examples**

**# block\_ip 192.168.1.50 1800 # Block for 30 minutes**

**# unblock\_ip 192.168.1.50**

**Lab 10: Troubleshooting and Best Practices**

**10.1 Debugging Rules**

**# Show rule handles**

**sudo nft --handle list table inet filter**

**# Show rule statistics**

**sudo nft --stateless list table inet filter**

**# Monitor traffic in real-time**

**sudo nft monitor**

**# Test rule matching**

**sudo nft add rule inet filter input tcp dport 22 counter accept**

**# Check counter values**

**sudo nft list table inet filter**

**10.2 Performance Optimization**

**# Use efficient rule ordering (most specific first)**

**sudo nft insert rule inet filter input position 1 ip saddr 192.168.1.0/24 accept**

**# Combine similar rules**

**sudo nft add rule inet filter input tcp dport { 80, 443, 8080 } accept**

**# Use maps for better performance**

**sudo nft add map inet filter port\_map { type inet\_service : verdict \; }**

**sudo nft add element inet filter port\_map { 22 : accept, 80 : accept, 443 : accept }**

**sudo nft add rule inet filter input tcp dport vmap @port\_map**

**10.3 Backup and Restore**

**# Backup current ruleset**

**sudo nft list ruleset > /tmp/nft-backup.nft**

**# Restore from backup**

**sudo nft -f /tmp/nft-backup.nft**

**# Atomic rule replacement**

**sudo nft -f /tmp/new-rules.nft**

**Practice Exercises**

**Exercise 1: Web Server Protection**

**Create a ruleset that:**

* **Allows SSH only from management network (192.168.100.0/24)**
* **Allows HTTP/HTTPS from anywhere**
* **Blocks more than 10 connections per minute from same IP**
* **Logs all blocked attempts**

**Exercise 2: Network Segmentation**

**Setup rules for:**

* **DMZ network (192.168.10.0/24) can access internet**
* **Internal network (192.168.20.0/24) can access DMZ and internet**
* **DMZ cannot access internal network**
* **Management network (192.168.100.0/24) can access everything**

**Exercise 3: Advanced NAT Gateway**

**Configure:**

* **Port forwarding for web server (80→192.168.1.100:80)**
* **SSH forwarding with port change (2222→192.168.1.100:22)**
* **Load balancing for multiple web servers**
* **SNAT for internal networks**

**Common Commands Reference**

**# List operations**

**sudo nft list tables**

**sudo nft list table inet filter**

**sudo nft list chain inet filter input**

**sudo nft --handle list table inet filter**

**# Rule operations**

**sudo nft add rule inet filter input tcp dport 22 accept**

**sudo nft insert rule inet filter input position 1 tcp dport 22 accept**

**sudo nft delete rule inet filter input handle 5**

**# Set operations**

**sudo nft add set inet filter myset { type ipv4\_addr \; }**

**sudo nft add element inet filter myset { 1.2.3.4 }**

**sudo nft delete element inet filter myset { 1.2.3.4 }**

**# Cleanup**

**sudo nft flush table inet filter**

**sudo nft delete table inet filter**

**sudo nft flush ruleset**

**Tips for Success**

1. **Always test rules carefully - Use nft add rule before making permanent changes**
2. **Keep backups - Save working configurations before modifications**
3. **Use meaningful names - Name tables, chains, and sets descriptively**
4. **Document your rules - Add comments explaining complex rules**
5. **Test connectivity - Always ensure you can still access the system**
6. **Monitor performance - Check rule statistics and optimize as needed**
7. **Use atomic operations - Replace entire rulesets rather than individual rules when possible**

**Remember: Always have a way to regain access to your system if rules go wrong!**