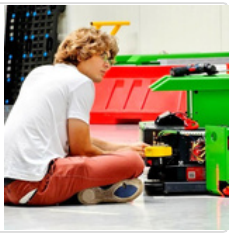
NET CS01: Master TOP 5 Networking skills and get Hired

Check GitHub for helpful DevOps tools:

Michael Robotics

Hi, I'm Michal. I'm a Robotics Engineer and DevOps enthusiast. My mission is to create skill-learning platform that combats information overload by adhering to the set of principles: simplify, prioritize, and execute.

https://github.com/MichaelRobotics



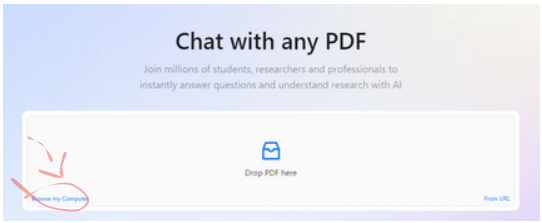
Ask Personal AI Document assistant to learn interactively (FASTER)!

1. Download PDF
2. Go to website
3. Browse file
4. Chat with Document

1 https://github.com/MichaelRobotics/DevOpsTools/blob/main/OSImodel.pdf

Click there to go to ChatPdf website 2

3



Ask questions about document! 4

TOP 5: Firewalls

1) Why - Problem case

The IT company is experiencing a security breach where unauthorized external actors are gaining access to the internal network. This problem has resulted in compromised sensitive data, disrupted operations.

2) Simple explanation

Employees and clients need to access certain resources remotely, such as web applications, email servers, and file sharing services. There's insufficient filtering of incoming and outgoing traffic, allowing potentially malicious traffic to enter the network. This includes lack of rules to block unwanted or harmful protocols and connections.

3) Solution

It professional after analysis of situation, decides that employes PC's have wrongly configured firewalls, So he decided to give them instructions for basic firewall settings, to block most of incoming traffic:

Install Firewall

```
$ sudo apt-get install ufw
```

Enable firewall

```
$ sudo ufw disable
```

Input settings recommended by professional

```
sudo ufw limit 22/tcp
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw enable
```

TOP 4: Distinguish IP addresses types

1) Why - Problem case

An IT company encountered frequent network outages and project delays when employees mistakenly assigned global unicast IPs to internal devices, causing IP conflicts, and used link-local addresses across network segments, leading to routing failures and disrupted client services.

2) Simple explanation

The issues occurred because global unicast IPs, intended for public internet use, were misapplied within the internal network, causing conflicts with public servers, while link-local addresses, meant for local segment communication, were incorrectly used for routing between different segments, leading to communication breakdowns.

3) Solution

To prevent such issues, the IT team should correctly assign private IP ranges for internal devices, use link-local addresses only within their intended segments, implement clear network management practices and train staff on proper IP address usage:

Global Unicast Addresses]

- Purpose: Used for unique, routable communication over the public internet.
- Example: 203.0.113.45

Unique Local Addresses (ULA)

- Purpose: Used for local communication within a private network or organization. They are not routable on the public internet.
- IPv4 Equivalent: Private address ranges like 10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16

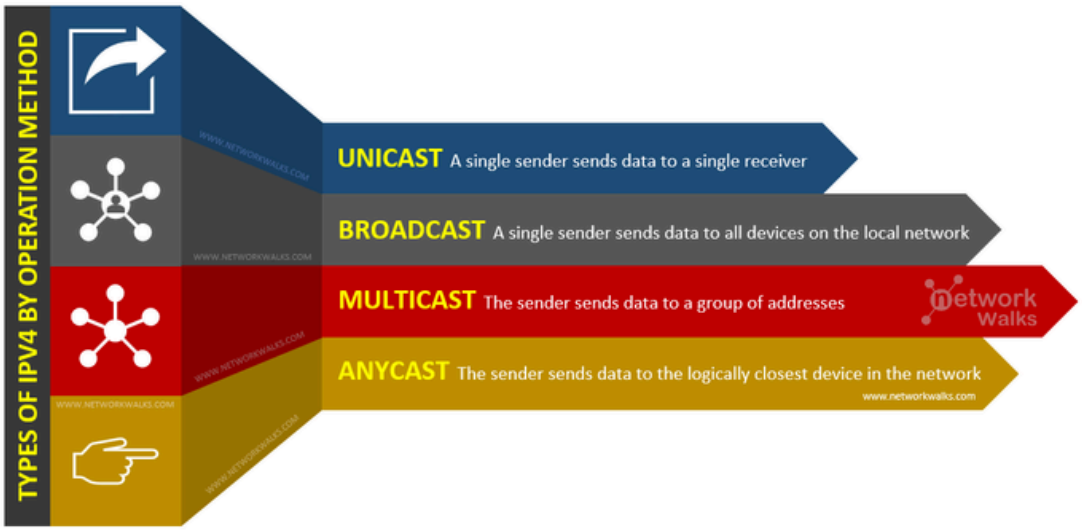
Multicast Addresses

- Purpose: Used to send data to multiple devices simultaneously. Devices join multicast groups to receive data sent to multicast addresses.
- Range: 224.0.0.0 to 239.255.255.255

6. **Broadcast Addresses**
   - Purpose: Used to send data to all devices on a specific network segment.
   - Characteristics: The highest address in a subnet (e.g., 192.168.1.255 for the 192.168.1.0/24 subnet).

7. **Loopback Addresses**
   - Purpose: Used by a device to communicate with itself. Useful for testing and troubleshooting.
   - Range: 127.0.0.0/8 (commonly 127.0.0.1).

For furher learning check: Anycast, Link-local, APIPA.



Link to great video about IPv4 and its addresses types:

What are the different IPv4 Address Types?
A quick overview of the different types of IPv4 addresses, their purpose and an example address

https://www.youtube.com/watch?v=xGONJA9saG8

TOP 3: Routing

1) Why - Problem Case

An IT company faced severe network disruptions when incorrect routing rules were set, causing packets to be improperly directed and resulting in failed communications between critical systems and significant downtime for client services.

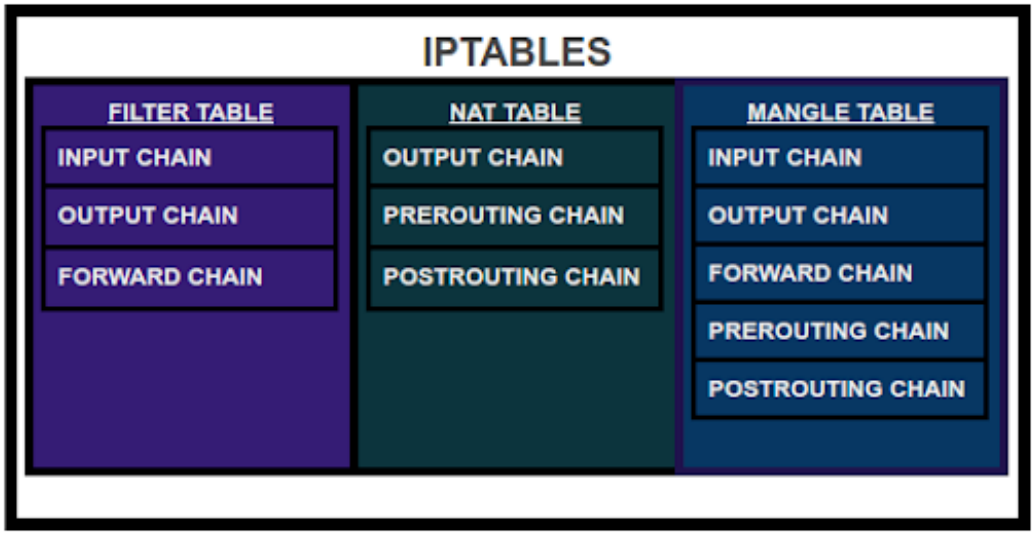
2) Simple Explanation

The disruptions happened because incorrect routing rules led to misdirected packets, which prevented proper communication between network systems and delayed application deployments.

3) Solution

To resolve these issues, the IT team should use Iptables for routing rules and receive training to implement and troubleshoot them effectively.

Iptables have 3 use cases: FILTER TABLE (firewall), MANGLE TABLE (modify packets) and NAT TABLE (routing). Understanding NAT TABLE module will help team to resolve issues:



1) The general syntax for iptables commands is:

```
$ iptables -t nat -A [CHAIN] -s [SOURCE] -d [DESTINATION] -p [PROTOCOL]
--dport [PORT] -j [TARGET]
```

-t nat specifies the NAT table.
-A [CHAIN] appends a rule to the specified chain.
-s [SOURCE] specifies the source IP address.
-d [DESTINATION] specifies the destination IP address.
-p [PROTOCOL] specifies the protocol (e.g., tcp, udp).
--dport [PORT] specifies the destination port.
-j [TARGET] specifies the target action (e.g., MASQUERADE, DNAT, SNAT).

2) Common NAT Rules and Examples

IP Masquerading

IP masquerading allows multiple devices on a local network to share a single public IP address.

Example: Masquerade all outgoing traffic from the local network (192.168.1.0/24) to use the public IP address of the gateway.

```
$ iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j MASQUERADE
```

Here:

- -s 192.168.1.0/24 specifies the source network.
- -o eth0 specifies the outgoing network interface.

Redirecting Outgoing Traffic to a Local Proxy Server

Objective: Redirect all outgoing HTTP requests from the local machine to the local proxy server running on port 8080.

```
$ iptables -t nat -A OUTPUT -p tcp --dport 80 -j REDIRECT --to-port 8080
```

Port Forwarding (DNAT)

Port forwarding redirects incoming traffic on a specific port to another port or IP address.

Example: Forward incoming traffic on port 80 (HTTP) to an internal web server at 192.168.1.10 on port 80.

```
$ iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.1.10:80
```

Here:

- -p tcp specifies the TCP protocol.
- --dport 80 specifies the destination port.
- --to-destination 192.168.1.10:80 specifies the internal destination IP and port.

Port Redirection (REDIRECT)

Port redirection is similar to port forwarding but redirects traffic to the local machine.

Example: Redirect incoming traffic on port 8080 to port 80 on the local machine.

```
iptables -t nat -A PREROUTING -p tcp --dport 8080 -j REDIRECT --to-port 80
```

Here:

- --to-port 80 specifies the local port to redirect traffic to.

SNAT (Source NAT)

SNAT changes the source IP address of outgoing packets.

Example: Change the source IP of outgoing traffic from 192.168.1.0/24 to the public IP address 203.0.113.1.

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j SNAT --to 203.0.113.1
```

Here:

- --to 203.0.113.1 specifies the public IP address.

TOP 2: TCP and UDP

1) Case Study

An IT company experienced significant network issues when TCP and UDP traffic was mismanaged, resulting in failed connections and disrupted services. Incorrect handling of TCP sessions led to unreliable data transmission.

2) Simple Explanation

The network problems arose because TCP and UDP traffic was improperly managed, causing unreliable connections and packet loss, which affected the performance and reliability of essential services and applications.

3) Solution

To address these issues, the IT team should use Wireshark to analyze TCP and UDP packets, identify misconfigurations or anomalies, and adjust network settings to prevent packet loss. The team will also need Wireshark training to effectively use the tool and interpret data accurately.

If you don't fully understand TCP and UDP, check out this video.

TCP vs UDP Comparison | Cisco CCNA 200-301
A network application has to choose how to send its data. That choice comes down to reliable or unreliable transmission.

https://www.youtube.com/watch?v=cA9ZJdqzOoU&t=185s

And best Wireshark explanation for TCP & UDP on yt:

Observing a TCP conversation in Wireshark
Using Wireshark, follow a TCP conversation, including 3-way handshake, sequence numbers and acknowledgements during an HTTP web request.

https://www.youtube.com/watch?v=cA9ZJdqzOoU&t=185s

TOP 1: ARP

1) Case Study:

A network loop in the Layer 2 network causes a broadcast storm, overwhelming the network with traffic and significantly reducing performance.

2) Simple Explanation:

The network is flooded with continuous broadcast messages, particularly ARP requests, leading to slowdowns or disruptions in communication across all devices.

3) Solution:

Implement Spanning Tree Protocol (STP) to eliminate loops and control broadcast traffic. Additionally, use tools like ARP inspection.

Develop great understanding of ARP:

ARP Explained | Address Resolution Protocol
ARP stands for Address Resolution Protocol. It's designed to discover MAC addresses and then map them to an IP address.

https://www.youtube.com/watch?v=tXzKjtMHgWI&t=4s

After understanding how ARP works, use those tools to troubleshoot:

arping

```
$ arping <IP address>
```

arping can be used to send ARP requests and measure the response time, which can help in diagnosing ARP-related issues.

traceroute

```
$ traceroute <IP address>
```

This tool helps trace the path packets take to reach a destination, which can be useful in identifying where packets might be getting lost.

tcpdump

```
$ sudo tcpdump -i <interface> arp
```

Use tcpdump to capture and analyze ARP packets. This can help identify if ARP requests and replies are being sent and received as expected.

arp

View ARP Table:

```
$ arp -n or ip neighbor
```

Add Entry:

```
$ sudo arp -s <IP address> <MAC address>
```

Delete Entry:

```
$ sudo arp -d <IP address>
```

Learn more about networking

Check HTB, they have great content

HTB - Your Cyber Performance Center

We provide a human-first platform creating and maintaining high performing cybersecurity individuals and organizations.

https://www.hackthebox.com/

Share, comment, DM and check GitHub for scripts & playbooks created to automate process.

Check my GitHub

Michael Robotics

Hi, I'm Michal. I'm a Robotics Engineer and DevOps enthusiast. My mission is to create skill-learning platform that combats skill information overload by adhering to the set of principles: simplify, prioritize, and execute.

https://github.com/MichaelRobotics

PS.

If you need a playbook or bash script to install KVM on a specific Linux distribution, feel free to ask me in the comments or send a direct message!