Linux Network Troubleshooting Commands

Traceroute and Tcpdump

Understand with Real Life Examples

DigiTalk

https://digitalksystems.com/

Reach us at digitalk.fmw@gmail.com
DigiTalk Channel: https://www.youtube.com/channel/UCCGTnI9vvF_ETMhGUXGdFWw
Playlists: https://www.youtube.com/@digitalk.middleware/playlists
Weblogic Server Architecture: https://youtu.be/gNqeIfLjUqw

DigiTalk Udemy Courses and Coupon Code

SOA Suite Administration
https://www.udemy.com/course/mastering-oracle-soa-suite-12c-administration/?couponCode=739A60915F86847014EB
Coupon Code: 739A60915F86847014EB

JBoss 8 Administration
https://www.udemy.com/course/mastering-jboss-eap-8-administration-from-intro-to-advanced/?couponCode=BF65EB008CFE16686BD2
Coupon Code:BF65EB008CFE16686BD2

OHS Administration
https://www.udemy.com/course/mastering-oracle-ohs-http-12c-web-server-administration/?couponCode=8E990556B21AF3E1A316
Coupon Code: 8E990556B21AF3E1A316

Weblogic Server Administration
https://www.udemy.com/course/oracle-weblogic-server-12c-and-14c-administration/?couponCode=87BC1314AC7690FD5294
Coupon Code:87BC1314AC7690FD5294

You can write us on digitalk.fmw@gmail.com if coupon code expired.

Traceroute

traceroute is a network diagnostic tool used to track the path that a packet takes from the source to the destination across an IP network. It helps identify the route and measure the transit delays of packets.

Understanding Delays in Traceroute Output

Each Hop's Response Time:

- Traceroute measures the time it takes for packets to reach each hop and receive a response.
- **Example Output:**

```
1  router.local (192.168.1.1)  1.234 ms  1.345 ms  1.456 ms
2  10.0.0.1 (10.0.0.1)  10.567 ms  10.678 ms  10.789 ms
3  203.0.113.1 (203.0.113.1)  25.678 ms  25.789 ms  25.890 ms
4  198.51.100.1 (198.51.100.1)  35.567 ms  35.678 ms  35.789 ms
5  example.com (93.184.216.34)  50.567 ms  50.678 ms  50.789 ms
```

- **Analysis:** Each line shows the response time from a hop. A significant increase in response time between hops indicates where delays are occurring.

Identifying Where Delays Occur:

- **Early Hops:** If delays are noticeable early in the traceroute (e.g., at hop 2 or 3), it suggests that there might be issues with your local network or ISP.
- **Middle Hops:** Delays in the middle of the path (e.g., hop 3 to 4) indicate problems with routing or congestion in the network between your ISP and the destination.
- **Final Hops:** Delays closer to the destination (e.g., hop 5) can suggest issues with the destination server or its network.

Real-Life Example

Imagine you're sending a package from your home to a friend's house, but it's arriving late. You use traceroute to track the route and see the following:

```
1  home-router.local (192.168.1.1)  2 ms  3 ms  2 ms
2  isp-router (10.0.0.1)  5 ms  6 ms  5 ms
3  regional-router (203.0.113.1)  20 ms  21 ms  22 ms
4  city-router (198.51.100.1)  50 ms  52 ms  51 ms
5  destination-server (93.184.216.34)  100 ms  98 ms  101 ms
```

Analysis:

- **Hops 1 and 2:** Fast response times suggest that there are no issues with your local network or ISP.
- **Hop 3:** Response time is higher than previous hops, indicating potential congestion or delay at the regional router.
- **Hop 4:** Response time increases significantly, showing potential problems with the city router or a network bottleneck in that area.
- **Hop 5:** Response time to the destination server is also high, suggesting possible issues with the destination server or its network.

Key Considerations

- **Network Congestion:** High latency at any hop can indicate congestion or traffic overload on that network segment.
- **Routing Issues:** Significant delays between hops can point to inefficient routing or suboptimal network paths.
- **Device Performance:** High response times might be due to performance issues with the routers or network devices themselves.

By analyzing where the delays occur in a traceroute output, you can pinpoint whether the issue is within your own network, with your ISP, or somewhere further along the path to the destination.

Usage:

traceroute [options] <destination>

Common Options:

- -m <max_ttl>: Set the maximum number of hops (TTL) to search for the destination.
- -p <port>: Specify the destination port to use.
- -q <nqueries>: Number of queries per hop (default is 3).
- -w <timeout>: Set the timeout for each reply.
- -I: Use ICMP ECHO instead of UDP datagrams (default is UDP).

Examples:

Basic Usage

Track the route to example.com:

traceroute example.com

Output:

traceroute to example.com (93.184.216.34), 30 hops max, 60 byte packets

```
 1  router.local (192.168.1.1)  1.234 ms  1.345 ms  1.456 ms
 2  10.0.0.1 (10.0.0.1)  10.567 ms  10.678 ms  10.789 ms
 ...
```

Set Maximum Number of Hops

Limit the trace to 10 hops:

```
traceroute -m 10 example.com
```

Use TCP Instead of UDP

Send TCP SYN packets to port 80:

```
traceroute -T -p 80 example.com
```

Specify Timeout

Set a 2-second timeout for each reply:

```
traceroute -w 2 example.com
```

Tcpdump

tcpdump is a command-line packet analyzer used to capture and display network packets being transmitted or received over a network. It's useful for troubleshooting network issues and analyzing traffic.

Real-Life Example:

Scenario:

Imagine you're running a cafe with a public Wi-Fi network, and you notice that the internet is running slowly. You suspect there might be a problem with how traffic is being routed or with some devices on the network.

Using tcpdump:

You want to capture and analyze the network traffic to identify any potential issues. You run tcpdump on your network router or a computer connected to the network:

```
sudo tcpdump -i eth0 -w capture.pcap
```

Explanation:

- -i eth0: Specifies the network interface to capture traffic from (e.g., eth0).
- -w capture.pcap: Writes the captured packets to a file named capture.pcap for later analysis.

Example Output:

```
13:45:01.123456 IP 192.168.1.2.54321 > 192.168.1.1.80: Flags [S], seq 123456789, win 65535, length 0
13:45:01.123457 IP 192.168.1.1.80 > 192.168.1.2.54321: Flags [S.], seq 987654321, ack 123456790, win 65535, length 0
13:45:01.123458 IP 192.168.1.2.54321 > 192.168.1.1.80: Flags [P.], seq 1, ack 1, win 65535, length 10
```

Analysis:

- **Packet Details:** Each line represents a packet and shows details such as source and destination IP addresses, ports, flags, and sequence numbers.
- **Example Breakdown:**
  - Line 1: A packet from IP 192.168.1.2 to 192.168.1.1 on port 54321 requesting to open a connection (SYN flag).
  - Line 2: The server (192.168.1.1) responds with an acknowledgment (SYN-ACK flag) and includes its own sequence number.
  - Line 3: The client (192.168.1.2) sends data to the server with a payload length of 10 bytes (PSH flag).

Key Considerations for Analysis

- **Filtering Traffic:** To focus on specific traffic, you can use filters. For example, to capture only HTTP traffic:
  sudo tcpdump -i eth0 port 80 -w http_traffic.pcap
- **Understanding Protocols:** tcpdump shows details of protocols like TCP, UDP, ICMP, etc. Understanding these protocols helps in interpreting the captured data.
- **Network Issues:** High volumes of retransmissions, unusual delays, or unexpected traffic patterns can indicate network problems such as congestion or faulty devices.
- **Security:** tcpdump can also help in identifying unauthorized access or malicious activity by analyzing unexpected or suspicious traffic patterns.

By capturing and analyzing network traffic with tcpdump, you can diagnose problems, optimize performance, and ensure the security of your network.

Usage:

tcpdump [options] [filter expression]

Common Options:

- -i <interface>: Specify the network interface to listen on (e.g., eth0, wlan0).
- -n: Show IP addresses and ports in numeric format, avoiding DNS lookups.
- -v: Verbose output.
- -c <count>: Number of packets to capture.
- -w <file>: Write captured packets to a file.
- -r <file>: Read packets from a file.

Examples:

Capture Packets on an Interface

Capture packets on eth0:

tcpdump -i eth0

Capture Specific Number of Packets

Capture 10 packets:

```
tcpdump -i eth0 -c 10
```

Capture Packets with a Specific Port

Capture packets on port 80 (HTTP):

```
tcpdump -i eth0 port 80
```

Capture Packets and Save to a File

Save captured packets to capture.pcap:

```
tcpdump -i eth0 -w capture.pcap
```

Read Packets from a File

Read packets from a file and display:

```
tcpdump -r capture.pcap
```

Filter by Protocol

Capture only TCP packets:

```
tcpdump -i eth0 tcp
```

Filter by IP Address

Capture packets to or from 192.168.1.1:

```
tcpdump -i eth0 host 192.168.1.1
```

Capture and Display in Human-Readable Format

Display packet content in a human-readable format:

```
tcpdump -i eth0 -X
```

Output:

```
14:31:12.123456 IP 192.168.1.2.12345 > 192.168.1.1.80: Flags [P], seq 12345:12378, ack 12345, win 1234, length 33
    0x0000: 4500 003d 1d4c 4000 4006 b1e6 c0a8 0102  E..=.L@.@.. ...
    0x0010: c0a8 0101 3039 0050 3001 7b1d 0000 0000  ....09.P0.{.....
```

Summary

Traceroute: Tracks the path packets take to reach a destination and measures transit delays. Useful for identifying network bottlenecks.

Tcpdump: Captures and displays network packets for analysis. Useful for detailed network traffic analysis and debugging.

Important considerations while doing the output analysis during troubleshooting

When analyzing the output from traceroute and tcpdump during network troubleshooting, there are several important considerations to keep in mind:

Traceroute

Considerations for Output Analysis:

Hop Count:

- What to Check: Ensure that the number of hops to the destination is reasonable.
- Why It Matters: A significantly higher number of hops may indicate routing inefficiencies or network problems.

Latency:

- What to Check: Look for high latency or timeouts in the response times.
- Why It Matters: High latency may indicate network congestion or issues with specific network segments.

Timeouts:

- What to Check: Note any timeouts or * * * entries in the trace.
- Why It Matters: Timeouts can indicate that a hop is not responding or that there is a problem with the network path.

IP Address Consistency:

- What to Check: Ensure that IP addresses are consistent and that routes are logical.
- Why It Matters: Unexpected IP changes or illogical routing may indicate network configuration issues.

Network Devices:

- What to Check: Identify the network devices along the path (routers, switches).
- Why It Matters: Understanding the network devices can help pinpoint where delays or problems are occurring.

Examples:

High Latency Example:

1 router.local (192.168.1.1) 1.234 ms 1.345 ms 1.456 ms

2 10.0.0.1 (10.0.0.1) 10.567 ms 10.678 ms 10.789 ms

3 203.0.113.1 (203.0.113.1) 100.567 ms 100.678 ms 100.789 ms

Analysis: Latency increases significantly after the second hop, indicating a potential issue with the network segment or device at hop 3.

Timeout Example:

1 router.local (192.168.1.1) 1.234 ms 1.345 ms 1.456 ms

2 10.0.0.1 (10.0.0.1) 10.567 ms 10.678 ms 10.789 ms

3 * * * Request timed out.

Analysis: The timeout at hop 3 may indicate a firewall or filtering issue, or that the device is not responding to traceroute requests.

Tcpdump

Considerations for Output Analysis:

Packet Capture Filters:

- What to Check: Use appropriate filters to capture relevant packets.
- Why It Matters: Capturing all packets without filters can lead to large amounts of data and make analysis difficult.

Packet Details:

- What to Check: Analyze packet headers, payloads, and flags for anomalies.
- Why It Matters: Detailed packet analysis helps in understanding communication issues, protocol errors, or malicious activity.

Protocol Analysis:

- What to Check: Identify and analyze different protocols (e.g., TCP, UDP, ICMP).
- Why It Matters: Problems with specific protocols can point to issues in application communication or network services.

Traffic Patterns:

- What to Check: Observe the traffic patterns, such as frequency and volume of packets.
- Why It Matters: Abnormal traffic patterns may indicate network congestion, attacks, or misconfigurations.

Error Messages:

- What to Check: Look for TCP flags (e.g., RST, SYN, FIN) and error messages.
- Why It Matters: Error messages and flags can provide insight into connection issues or failures.

Examples:

Basic Packet Capture:

tcpdump -i eth0

Analysis: Monitor all packets to observe general traffic and identify unusual patterns.

Filter by Port:

tcpdump -i eth0 port 80

Analysis: Capture HTTP traffic to analyze web server interactions and potential issues.

Analyze TCP Flags:

tcpdump -i eth0 'tcp[tcpflags] & (tcp-rst) != 0'

Analysis: Identify and investigate TCP RST (reset) flags, which may indicate connection issues.

Inspect Packet Payload:

tcpdump -i eth0 -X

Analysis: Examine packet contents to understand the data being transmitted and identify potential issues.

Summary

Traceroute:

- **Key Points:** Check hop count, latency, timeouts, IP address consistency, and network devices.
- **Use Case:** Identify routing issues and network bottlenecks.

Tcpdump:

- **Key Points:** Use appropriate filters, analyze packet details, observe protocol and traffic patterns, and check for error messages.
- **Use Case:** Diagnose network problems, analyze traffic, and detect anomalies.

DISCLAIMER AND CONSENT

This document is being provided by DigiTalk as part of its effort to assist users in understanding and working with Linux. While every effort has been made to ensure the accuracy and reliability of the information presented in this document, there is a possibility of typographical errors or inaccuracies. DigiTalk does not guarantee the correctness or completeness of the content provided in this document.

Users of this document are encouraged to cross-reference the information presented here with official documentation available on their website or other authoritative sources. Any discrepancies or inaccuracies found in this document should be reported to us at digitalk.fmw@gmail.com.

By using this document, you acknowledge and consent to the following:

This document is not officially endorsed or verified by any other third party organization..

The Company makes no claims or guarantees about the accuracy or suitability of the information contained in this document.

Users are responsible for verifying and validating any information presented here for their specific use case.

DigiTalk disclaims any liability for any errors, omissions, or damages that may result from the use of this document.

If you discover any inaccuracies or errors in this document, please report them to digitalk.fmw@gmail.com, and the Company will endeavor to correct them as necessary.

This consent statement is provided to ensure transparency and understanding of the limitations of the information contained in this document. By using this document, you agree to abide by the terms and conditions outlined herein.