Ethical Hacking using the Switch Port Analyser in a Enterprise Network

Charukesh,
Indian Institute of Information Technology Design and Manufacturing
Kurnool, Andhra Pradesh charukesh.charu@gmail.com

***Abstract*— SPAN (Switched Port Analyzer) is a network feature that allows network administrators to monitor network traffic by copying network packets from one port to another. In this research paper, we will discuss the basics of SPAN port and its functionalities. We will also cover how SPAN port works, its advantages, and limitations. Furthermore, we will discuss the different types of SPAN port, including local SPAN, remote SPAN, and RSPAN. Finally, we will also discuss the best practices for configuring and using SPAN ports.**

Keywords— SPAN Port, Packet Capture, Packet Monitor, Eavesdropping.

I. INTRODUCTION TO SPAN PORT

In today's world, where network security has become a top priority for organizations, monitoring network traffic has become crucial. Network administrators need to keep a close eye on their network traffic to detect any suspicious activities, troubleshoot network issues, and analyze network performance. SPAN port, also known as port mirroring, is a network feature that enables network administrators to monitor network traffic without disrupting the network. This research paper aims to provide a detailed analysis of SPAN port and its functionalities.

A. What is SPAN Port?

SPAN (Switched Port Analyzer) is a network feature that allows network administrators to copy network traffic from one port or VLAN and send it to another port for analysis. In simple words, it is a method of copying network traffic from one port to another without disrupting the flow of traffic. The copied traffic can be monitored, analyzed, or stored for later use.[1]

B. History of SPAN Port

The history of SPAN (Switched Port Analyzer) port goes back to the early days of network switches. In the early 2000s, as network switches became more prevalent and replaced traditional hubs, network administrators faced new challenges in monitoring and analyzing network traffic. Unlike hubs, which simply replicated all network traffic on all ports, switches used packet switching to direct traffic only to the intended destination port. This meant that network administrators needed a new way to monitor network traffic in real-time, especially for troubleshooting and security purposes.

The SPAN port was introduced as a solution to this problem. The first SPAN port was implemented in the Cisco Catalyst 5000 switch in 1996, which allowed network administrators to monitor network traffic by copying all traffic passing through the switch to a designated monitoring port. The SPAN port quickly became a popular feature on network switches, and other switch vendors began to offer their own versions of the feature.

Over the years, the SPAN port has evolved and become more sophisticated. Modern SPAN ports support a range of filtering options, such as filtering by port, VLAN, or protocol, to allow for more targeted monitoring of network traffic. Additionally, new monitoring tools have been developed that can analyze and visualize network traffic, providing valuable insights into network performance, security, and user behavior.

Today, the SPAN port remains a key feature of network switches and a valuable tool for network administrators and analysts to monitor and analyze network traffic in real-time.

II. HOW DOES OF SPAN PORT WORK, ADVANTAGES AND DISADVANTAGES AND ITS TYPES

A SPAN (Switched Port Analyzer) port is a feature available on many network switches that allows network administrators to monitor network traffic in real-time. The SPAN port copies all network traffic passing through the switch to a designated monitoring port, where the traffic can be captured and analyzed by monitoring tools such as packet analyzers, intrusion detection systems, or network performance monitors.

Here are some key details about the SPAN port:

- The SPAN port is also sometimes called a mirror port, monitoring port, or capture port.
- The SPAN port is a passive monitoring feature, meaning that it does not alter or affect the original network traffic.
- The SPAN port copies all traffic passing through the switch, including both incoming and outgoing traffic.
- The SPAN port can be configured to monitor traffic for specific ports, VLANs, or protocols using various filtering options.
- The SPAN port can be used for various network monitoring and analysis tasks, including security monitoring, troubleshooting network issues, analyzing network performance, and monitoring user activity.
- The SPAN port can be configured and managed through the switch's management interface or command-line interface.

Overall, the SPAN port is a valuable tool for network administrators and analysts to monitor and analyze network traffic in real-time, providing insights into network performance, security, and user behavior. However, it's important to use the SPAN port and associated monitoring tools in a responsible and ethical manner, and to ensure that

one will have proper authorization and consent to intercept and analyze network traffic.SPAN port works by copying the network packets from one port or VLAN and forwarding them to another port. The copied packets are then analyzed by a network analyzer or monitoring tool. The network analyzer or monitoring tool can be a software tool or a hardware device. SPAN port can be configured on a switch, router, or firewall, depending on the network infrastructure.[2]

A. Advantages of SPAN Port:

SPAN port has several advantages, some of which are:

1. Network Monitoring: SPAN port allows network administrators to monitor network traffic without disrupting the flow of traffic. It helps them to detect any suspicious activities, troubleshoot network issues, and analyze network performance.
2. Cost-Effective: SPAN port is a cost-effective solution for network monitoring as it does not require any additional hardware.
3. Traffic Analysis: SPAN port enables network administrators to analyze network traffic and identify the root cause of network issues.
4. Compliance: SPAN port helps organizations to comply with regulatory requirements such as HIPAA, PCI DSS, and SOX by monitoring network traffic.

B. Limitations of SPAN Port:

SPAN port has a few limitations, some of which are:

1. Limited Monitoring: SPAN port can only monitor the traffic that passes through the switch, and it cannot monitor traffic that bypasses the switch.
2. Performance Impact: SPAN port can impact the performance of the switch as it copies network packets, which can increase the CPU utilization of the switch.
3. Security Risks: SPAN port can pose security risks if not configured correctly. It can enable unauthorized access to network traffic.

C. Types of SPAN Port:

There are three types of SPAN port, which are:

1. Local SPAN: Local SPAN is used to copy network traffic from one port on the same switch and forward it to another port on the same switch.
2. Remote SPAN: Remote SPAN is used to copy network traffic from one port on one switch and forward it to another port on a different switch.
3. RSPAN: RSPAN (Remote Switched Port Analyzer) is used to copy network traffic from one or more ports on multiple switches and forward it to a monitoring port on a different switch.

D. Best Practices for Configuring and Using SPAN Port:

Here are some best practices for configuring and using SPAN port:

1. Configure SPAN port only when necessary.
2. Use a dedicated port for SPAN traffic to avoid any impact on regular traffic.
3. Limit the amount of traffic being copied to the monitoring port to avoid overwhelming the monitoring tool.
4. Configure access control lists (ACLs) to restrict access to the monitoring port.
5. Regularly review the SPAN port configuration to ensure it meets the organization's security policies.

III. CONFIGURING THE SPAN PORT

A. Best Practices for Configuring and Using SPAN Port

Here are some best practices for configuring and using SPAN port:

1. Configure SPAN port only when necessary.
2. Use a dedicated port for SPAN traffic to avoid any impact on regular traffic.
3. Limit the amount of traffic being copied to the monitoring port to avoid overwhelming the monitoring tool.
4. Configure access control lists (ACLs) to restrict access to the monitoring port.
5. Regularly review the SPAN port configuration to ensure it meets the organization's security policies.

B. Configuring a SPAN Port

Step 1: Determine the Traffic to be Monitored

The first step in configuring a SPAN port is to determine the traffic to be monitored. This can include traffic from a specific port or VLAN, or traffic from multiple ports or VLANs. It's important to identify the traffic that needs to be monitored to avoid overwhelming the monitoring tool.

Step 2: Identify the Destination Port

The next step is to identify the destination port where the monitored traffic will be sent. The destination port can be a physical port or a logical port. It's important to select a port that can handle the amount of traffic being monitored.

Step 3: Configure the SPAN Session

The next step is to configure the SPAN session. This involves specifying the source port or VLAN, the destination port, and any additional parameters such as VLAN tags or traffic filters. The configuration process can vary depending on the switch vendor and model.

Below is an Code Snippet for configuring a SPAN session on a Cisco switch:

Switch(config)# monitor session 1 source interface gigabitethernet0/1

Switch(config)# monitor session 1 destination interface gigabitethernet0/2

This configuration creates a SPAN session where traffic from GigabitEthernet0/1 is copied to GigabitEthernet0/2.

Step 4: Verify the SPAN Configuration

After configuring the SPAN session, it's important to verify that it's working as intended. This can be done by using

a network analyzer or monitoring tool to capture and analyze the monitored traffic. The tool should be able to capture the same traffic that is being sent to the destination port.[3]

C. Code Snippet for packet capturing using SPAN port in Python:

```
from scapy.all import *

# Define the interface to capture packets from (the SPAN port)
iface = "eth0"

# Define the filter to capture specific packets
filter = "tcp and port 80"

# Define the function to handle captured packets
def handle_packet(packet):
    # Process the packet here
    print(packet.summary())

# Start the packet capture
sniff(iface=iface, filter=filter, prn=handle_packet)
```

This code uses the **scapy** library to capture packets from the SPAN port on the **eth0** interface. It filters the captured packets to only include TCP packets with destination port 80 (HTTP traffic). The **handle_packet()** function is called for each captured packet, allowing to process and analyze the packets as needed. Finally, the **sniff()** function starts the packet capture process.

D. Code Snippet for retrieving data from captured packets using SPAN port in Python:

```
from scapy.all import *

# Define the interface to capture packets from (the SPAN port)
iface = "eth0"

# Define the filter to capture specific packets
filter = "tcp and port 80"

# Define the function to extract data from captured packets
def extract_data(packet):
    # Extract data from the packet here
    if packet.haslayer(TCP) and packet.haslayer(Raw):
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        src_port = packet[TCP].sport
        dst_port = packet[TCP].dport
        data = packet[Raw].load.decode('utf-8')
        print(f"Source IP: {src_ip} | Destination IP: {dst_ip} | Source Port: {src_port} | Destination Port: {dst_port} | Data: {data}")

# Start the packet capture
sniff(iface=iface, filter=filter, prn=extract_data)
```

This code uses the scapy library to capture packets from the SPAN port on the eth0 interface. It filters the captured packets to only include TCP packets with destination port 80 (HTTP traffic). The extract_data() function is called for each captured packet that contains both a TCP and Raw layer. It extracts the source IP, destination IP, source port, destination port, and data payload from the packet, and prints the information to the console. Finally, the sniff() function starts the packet capture process.

E. Code Snippet for Packet Monitor, Packet Capture and Packet Retrieval using SPAN Port

```
from scapy.all import *

# Define the source and destination interface for the SPAN port
src_interface = "eth0"
dst_interface = "eth1"

# Define the packet filter to capture only HTTP traffic
packet_filter = "tcp port 80"

# Define the file to store captured packets
capture_file = "captured_packets.pcap"

# Define a function to handle captured packets
def handle_packet(packet):
    # Print the packet details
    print(packet.summary())

    # Write the packet to the capture file
    wrpcap(capture_file, packet, append=True)

# Start capturing packets using the SPAN port sniff(iface=src_interface, filter=packet_filter, prn=handle_packet)

# Retrieve captured packets from the capture file
captured_packets = rdpcap(capture_file)

# Print the details of the captured packets
```

```
for packet in captured_packets:
    print(packet.summary())
```

In this Code, we first define the source and destination interface for the SPAN port, as well as the packet filter to capture only HTTP traffic. We then define a function to handle the captured packets, which simply prints the packet details and writes the packet to a capture file.

We start capturing packets using the **sniff()** function from the Scapy library, which captures packets from the specified interface and applies the packet filter. The **prn** parameter specifies the function to handle each captured packet.

After capturing packets, we retrieve them from the capture file using the **rdpcap()** function from the Scapy library. We then print the details of the captured packets using a for loop.

Note that this is just a simple example code, and there are many other parameters and options available for the **sniff()** and **rdpcap()** functions. Additionally, it's important to use SPAN port and packet capturing tools in a responsible and ethical manner, and to ensure that one will have proper authorization and consent to intercept and analyze network traffic.

F. Code Snippet for Capturing packets from all TCP Ports or connections using the SPAN Port

```
from scapy.all import *

# Define the source and destination interface for the SPAN port
src_interface = "eth0"
dst_interface = "eth1"

# Define the packet filter to capture all TCP traffic
packet_filter = "tcp"

# Define the file to store captured packets
capture_file = "captured_packets.pcap"

# Define a function to handle captured packets
def handle_packet(packet):
    # Print the packet details
    print(packet.summary())

    # Write the packet to the capture file
    wrpcap(capture_file, packet, append=True)

# Start capturing packets using the SPAN port sniff(iface=src_interface, filter=packet_filter, prn=handle_packet)

# Retrieve captured packets from the capture file
captured_packets = rdpcap(capture_file)

# Print the details of the captured packets
for packet in captured_packets:
    print(packet.summary())
```

In this Code Snippet , we use the packet filter "tcp" to capture all TCP traffic, regardless of the source or destination port. This will capture packets from all TCP ports or connections on the network.

We start capturing packets using the **sniff()** function from the Scapy library, which captures packets from the specified interface and applies the packet filter. The **prn** parameter specifies the function to handle each captured packet.

After capturing packets, we retrieve them from the capture file using the **rdpcap()** function from the Scapy library. We then print the details of the captured packets using a for loop.

Note that capturing packets from all TCP ports or connections can generate a large amount of traffic and may not be practical for long-term monitoring or analysis. It's important to use appropriate filtering and analysis techniques to focus on specific traffic of interest and to avoid overwhelming the capture device or the network.

Also note that one will need to have root or administrator privileges to capture packets from the SPAN port. Additionally, the specific interface and filter used in the code will need to be customized for the network setup and monitoring needs.

G. Intergrating the SPAN Port and Wireshark for packet capturing

Integrating the SPAN port and Wireshark for packet capturing can be useful for analyzing network traffic and troubleshooting network issues. Here are the steps to configure Wireshark to capture packets from a SPAN port:

1. Connect the SPAN port to the computer running Wireshark.
2. Open Wireshark and go to Capture > Options.
3. Select the interface that corresponds to the SPAN port from the list of available interfaces.
4. Set any desired capture filters in the "Capture Filter" field to limit the packets captured to a specific protocol or traffic type.
5. Click the "Start" button to begin capturing packets from the SPAN port.
6. After capturing packets, one can view the details of each packet in the packet list, and use various analysis tools in Wireshark to analyze the traffic.[10]

Here is an example of packet tracing results in Wireshark

```
Length  Info
1    0.000000000      192.168.0.1
192.168.0.2            TCP        66
```

```
80 → 54321 [SYN, ACK] Seq=0 Ack=1
Win=29200 Len=0 MSS=1460 SACK_PERM=1

2    0.000015000     192.168.0.2
192.168.0.1           TCP       54
54321 → 80 [ACK] Seq=1 Ack=1 Win=29200
Len=0

3    0.005781000     192.168.0.2
192.168.0.1           HTTP      273
GET /index.html HTTP/1.1

4    0.005797000     192.168.0.1
192.168.0.2           TCP       54
80 → 54321 [ACK] Seq=1 Ack=220 Win=29200
Len=0

5    0.005843000     192.168.0.1
192.168.0.2           HTTP      301
HTTP/1.1 200 OK  (text/html)

6    0.005857000     192.168.0.2
192.168.0.1           TCP       54
54321 → 80 [ACK] Seq=220 Ack=248
Win=29200 Len=0
```

IV. ALTERNATIVES FOR THE SPAN PORT

While the SPAN (Switched Port Analyzer) port is a useful tool for monitoring network traffic, there are several alternatives

hat can be used in different scenarios. Here are some of the alternatives to the SPAN port:

1. TAP (Test Access Point): A TAP is a dedicated hardware device that copies all network traffic passing through it to a monitoring port. Unlike the SPAN port, which is a feature built into network switches, a TAP is a separate device that sits between two network devices and passively copies all traffic. TAPs are often used in high-performance environments where there is a need for full-duplex monitoring of network traffic.
2. Inline network security devices: Inline network security devices, such as firewalls and intrusion prevention systems (IPS), sit between network devices and actively inspect and filter traffic as it passes through. These devices are often used to monitor and block malicious traffic in real-time.
3. Protocol analyzers: Protocol analyzers, also known as packet sniffers, are software applications that capture and analyze network traffic. They are typically installed on a host device and capture traffic from the network interface of that device. Protocol analyzers can be used for a variety of tasks, such as troubleshooting network issues, analyzing network performance, and monitoring user activity.
4. NetFlow: NetFlow is a protocol developed by Cisco that captures and exports network traffic statistics from network devices, such as routers and switches. NetFlow can be used to monitor traffic patterns, identify network bottlenecks, and detect unusual traffic behavior.[4]

Overall, the choice of monitoring tool depends on the specific requirements of the network environment and the goals of the monitoring task. While the SPAN port is a common and useful tool for monitoring network traffic, alternatives such as TAPs, inline security devices, protocol analyzers, and NetFlow can also be effective in certain situations.

V. CONCLUSION

SPAN (Switched Port Analyzer) ports are a valuable tool for network administrators and analysts to monitor and analyze network traffic in real-time. However, it is important to use SPAN ports ethically and in compliance with applicable laws and regulations. The use of SPAN ports should be restricted to authorized personnel who have a legitimate need to monitor and analyze network traffic for security, performance, or troubleshooting purposes. Unauthorized monitoring of network traffic can violate privacy laws and may lead to legal consequences.

Furthermore, network administrators should be transparent about the use of SPAN ports and inform users that their network activity may be monitored. It is also important to implement appropriate security measures, such as encryption, to protect sensitive information that may be captured through SPAN ports.

Ultimately, the responsible use of SPAN ports requires a balance between the need to monitor network traffic for legitimate purposes and the protection of individual privacy and data security. By following ethical and legal guidelines, SPAN ports can be a powerful tool for maintaining a secure and reliable network infrastructure.

REFERENCES

[1] Farrokhi, S., & Hedayati, A. (2021). A Security Solution for Network Threats Detection Based on SPAN Port. IEEE Access, 9, 38524-38534.

[2] Islam, M. M., & Yeun, C. Y. (2021). Analysis of Network Traffic Monitoring in Cloud Computing Environments. IEEE Access, 9, 133047-133063.

[3] Kim, D. H., & Cho, Y. (2020). A monitoring framework of distributed servers using switched port analyzer (SPAN) in cloud environment. Journal of Supercomputing, 76(7), 5149-5165.

[4] Peng, Y., Zeng, L., Li, Z., & Zhao, H. (2020). A QoS-Aware Traffic Sampling Approach for Monitoring SDN Networks. IEEE Access, 8, 146950-146960.

[5] Su, S., Li, Z., Zhang, X., Yu, L., & Li, G. (2019). Improved traffic classification method based on machine learning algorithms. Multimedia Tools and Applications, 78(17), 24881-24897.

[6] Cisco. (n.d.). Configuring S P A N and RSPAN. Retrieved from https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4000/8-2glx/configuration/guide/span.html

[7] Cisco. (n.d.). Configuring VLAN Access Maps. Retrieved from https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus9000/sw/7-x/security/configuration/guide/b_Cisco_Nexus_9000_Series_NX-OS_Security_Configuration_Guide_7x/b_Cisco_Nexus_9000_Series_NX-OS_Security_Configuration_Guide_7x_chapter_01011.html

[8] Cisco. (n.d.). SPAN on Cisco IOS Software-Based Catalyst Switches Configuration Example. Retrieved from https://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/10570-41.html

[9] Gibson, D. (2018). CompTIA Security+ Get Certified Get Ahead: SY0-501 Study Guide. Virginia Beach, VA: YCDA, LLC.

[10] Wireshark. (n.d.). Capturing on Switched Networks. Retrieved from https://osqa-ask.wireshark.org/questions/20328/how-do-i-capture-all-traffic-on-a-switch/