StrongSwan VPN

Project Overview

This project focused on designing and implementing a **secure IPsec VPN tunnel** using **StrongSwan** between two virtual machines in an enterprise-style network. The VPN provided encrypted communication between internal and DMZ environments, ensuring confidentiality and integrity of data in transit.

Additionally, the VPN server was integrated with **Splunk Enterprise** to collect, monitor, and analyze VPN logs, providing real-time visibility into connection attempts, authentications, and potential anomalies.

This project demonstrates practical skills in **secure network configuration**, **firewall and routing management**, and **SIEM integration** for log visibility and threat detection.

Objectives

- Configure **StrongSwan IPsec VPN** for secure host-to-host communication.
- Assign and manage static IP addresses for VPN endpoints.
- Update DNS configurations and DHCP services for proper routing.
- Forward VPN logs to **Splunk Enterprise** for monitoring.
- Validate encryption through **Wireshark** traffic analysis.
- Apply **UFW firewall rules** to restrict SSH access within the DMZ.

Implementation Steps

1. Lab Preparation and Environment Setup

- Cloned the existing *Client2-DMZ* virtual machine to create *StrongSwan-DMZ*.
- Enabled “Reinitialize MAC addresses” during cloning to prevent conflicts.
- Verified network connectivity and booted the **Outside-Firewall** before powering on *StrongSwan-DMZ*.

---

2. Hostname and Static IP Configuration

- Changed the hostname from `client2-dmz` to `strongswan-dmz`:
- `sudo nano /etc/hostname`
- `sudo nano /etc/hosts`
- `sudo hostname strongswan-dmz`
- Assigned a static IP (192.168.4.60) under the **IPv4 Settings** tab using "Manual" configuration.
- Set **DNS Server** to `192.168.4.1` to align with internal DNS routing.
- Restarted the network interface to apply changes:
- `sudo ifconfig enp0s3 down`
- `sudo ifconfig enp0s3 up`

---

3. Fixing DNS Configuration and DHCP Server

- Updated **pfSense DHCP settings** on *Inside-Firewall* to distribute the correct DNS server (`192.168.4.1`).
- Navigated to:
  Services → DHCP Server → Servers
  - Replaced Google DNS (`8.8.8.8`) with local DNS (`192.168.4.1`).
  - Saved and applied changes.
- Confirmed that *Inside-Firewall* and *Outside-Firewall* continued to route internal traffic correctly.

---

4. Installing and Configuring StrongSwan VPN Server

- Installed required StrongSwan packages:
- `sudo apt update`
- `sudo apt install strongswan strongswan-pki -y`
- Configured `/etc/ipsec.conf`:(sudo nano /etc/ipsec.conf)
- `config setup`
- `conn %default`
- `    ikelifetime=60m`
- `    keylife=20m`
- `    rekeymargin=3m`
- `    keyingtries=1`
- `    keyexchange=ikev2`
- `    authby=secret`
- `    mobike=no`
-
- `conn host-host`
- `    keyexchange=ikev2`
- `    ike=aes256-sha384-ecp384-prfsha384!`
- `    esp=aes256gcm128-ecp384!`
- `    left=%any`
- `    leftfirewall=yes`

- `right=192.168.4.60`
- `type=transport`
- `auto=add`
- Defined pre-shared keys in `/etc/ipsec.secrets`:
- `%any : PSK "password"`
- Restarted the StrongSwan service:
- `sudo systemctl restart strongswan-starter`
- `sudo ipsec statusall`

---

5. Enabling IP Forwarding and Security Controls

- Modified `/etc/sysctl.conf` to enable routing and disable redirects:
- `$ sudo su -`
- `# echo "net.ipv4.ip_forward = 1" | tee -a /etc/sysctl.conf`
- `# echo "net.ipv4.conf.all.accept_redirects = 0" | tee -a /etc/sysctl.conf`
- `# echo "net.ipv4.conf.all.send_redirects = 0" | tee -a /etc/sysctl.conf`
- `# echo "net.ipv4.conf.default.rp_filter = 0" | tee -a /etc/sysctl.conf`
- `# echo "net.ipv4.conf.default.accept_source_route = 0" | tee -a /etc/sysctl.conf`
- `# echo "net.ipv4.conf.default.send_redirects = 0" | tee -a /etc/sysctl.conf`
- `# echo "net.ipv4.icmp_ignore_bogus_error_responses = 1" | tee -a /etc/sysctl.conf`
- `# sysctl -p`
- `# exit`
- Applied changes using:

---

6. Integrating StrongSwan with Splunk

Objective: Forward VPN logs to Splunk for real-time visibility.

- Modified the StrongSwan logging configuration file `/etc/strongswan.d/charon-logging.conf`:
- `charon {`
- `    filelog {`
- `        charon {`
- `            path = /var/log/charon.log`
- `            time_format = %b %e %T`
- `            ike_name = yes`
- `            append = no`
- `            default = 2`
- `            flush_line = yes`
- `        }`

```
        stderr {
            ike = 2
            knl = 3
        }
    }

    syslog {
        identifier = charon-custom
        daemon {
        }
        auth {
            default = -1
            ike = 0
        }
    }
}
```

- Updated **AppArmor** permissions:

```
/var/log/ r,
/var/log/** rwk,
```

Then restarted AppArmor:

```bash
sudo service apparmor restart
```

---

7. Configuring Splunk Enterprise Receiver

- On the **Splunk-DMZ** machine (192.168.4.20):
  - **Settings → Forwarding and Receiving → New Receiving Port (9997)**
  - Created a new **index** named `strongswan` for VPN logs.

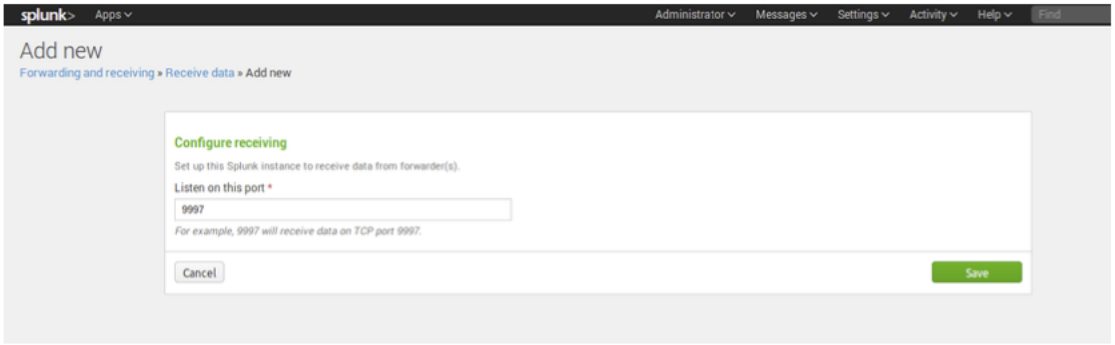

Figure 1/Splunk

---

8. Installing Splunk Forwarder on StrongSwan-DMZ

- Installed Splunk Universal Forwarder and enabled boot-start:

- `sudo dpkg -i splunkforwarder.deb`
- `sudo /opt/splunkforwarder/bin/splunk enable boot-start`
- `sudo service SplunkForwarder start`
- Added Splunk server and log path:
- `sudo /opt/splunkforwarder/bin/splunk add forward-server 192.168.4.20:9997`
- `sudo /opt/splunkforwarder/bin/splunk add monitor /var/log/charon.log -index strongswan`

---

9. Configuring the VPN Client (Client2-DMZ)

- Installed StrongSwan and configured `/etc/ipsec.conf`:
- `config setup`
-
- `conn %default`
- `ikelifetime=60m`
- `keylife=20m`
- `rekeymargin=3m`
- `keyingtries=1`
- `keyexchange=ikev2`
- `authby=secret`
- `mobike=no`
-
- `conn host-host`
- `keyexchange=ikev2`
- `ike=aes256-sha384-ecp384-prfsha384!`
- `esp=aes256gcm128-ecp384!`
- `left=%any`
- `leftfirewall=yes`
- `right=192.168.4.60`
- `auto=start`
- `type=transport` Defined shared key in `/etc/ipsec.secrets`:
- `192.168.4.60 : PSK "password"`
- Restarted VPN service:
- `sudo systemctl restart strongswan-starter`
- `sudo ipsec statusall`

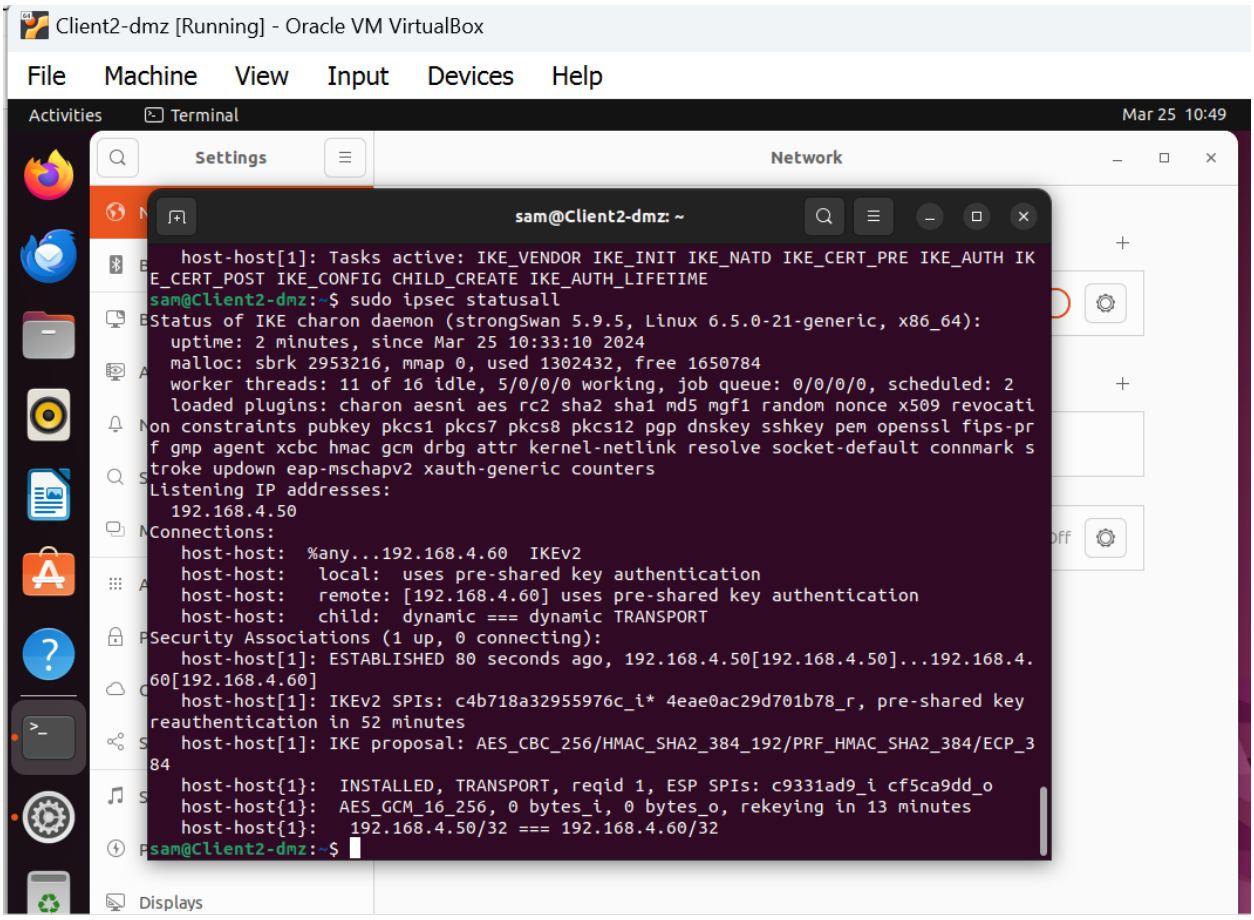

Figure 2/Statusall

---

10. Testing and Verification

- Verified the tunnel by running:
- `ping 192.168.4.60`
- Captured traffic using **Wireshark** — confirmed encrypted ESP packets.
- Disconnected VPN and confirmed communication dropped as expected.
- Verified successful log forwarding in **Splunk** using queries:
- `index=strongswan`
- `source="/var/log/charon.log"`
- Logs showed successful IKE negotiations, rekey events, and connection teardowns.

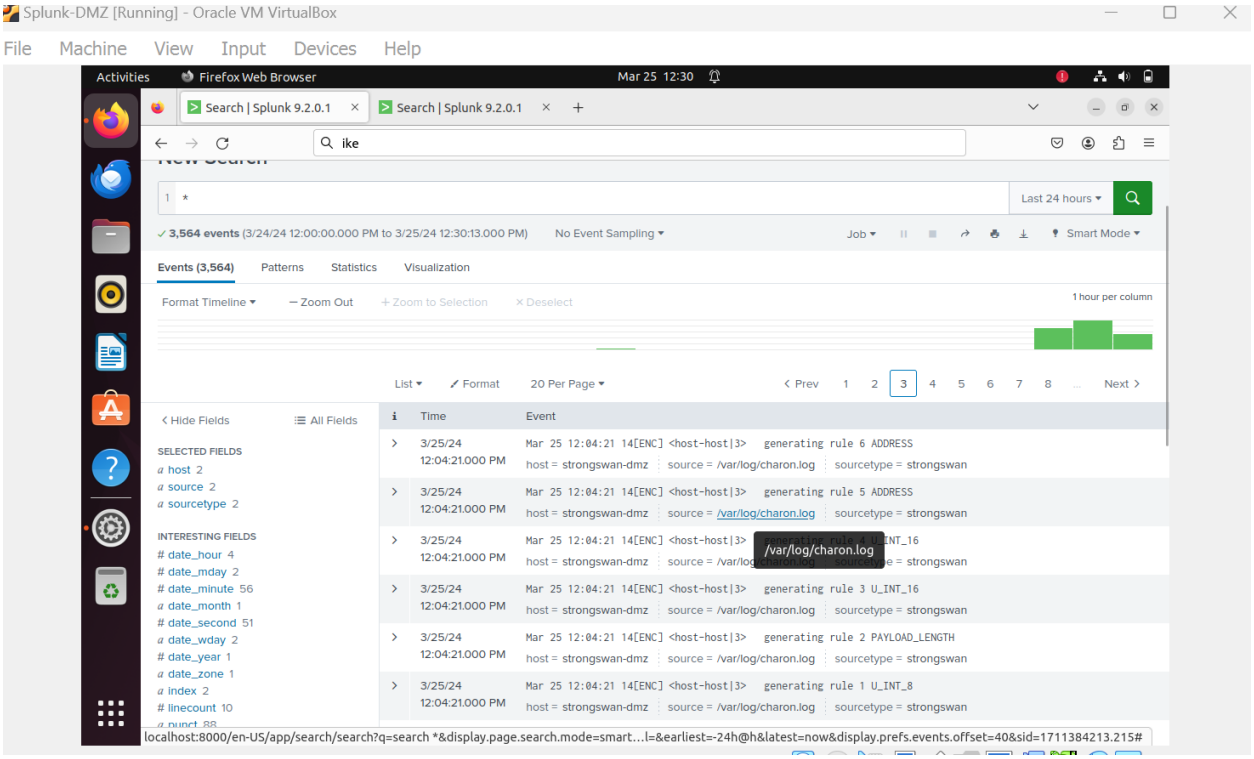

Figure 3/Splunk