Application Layer Security

CYBR3010

Professor: Sam El’Awour

Gregory Stephens

November 8, 2025

Introduction

This lab demonstrates the configuration and verification of **SSL/TLS Deep Inspection** on a FortiGate virtual firewall within a **Cisco Modeling Labs (CML)** environment. The objective was to decrypt, inspect, and re-encrypt HTTPS traffic traversing the network to understand how encrypted communications can be securely analyzed at the application layer.

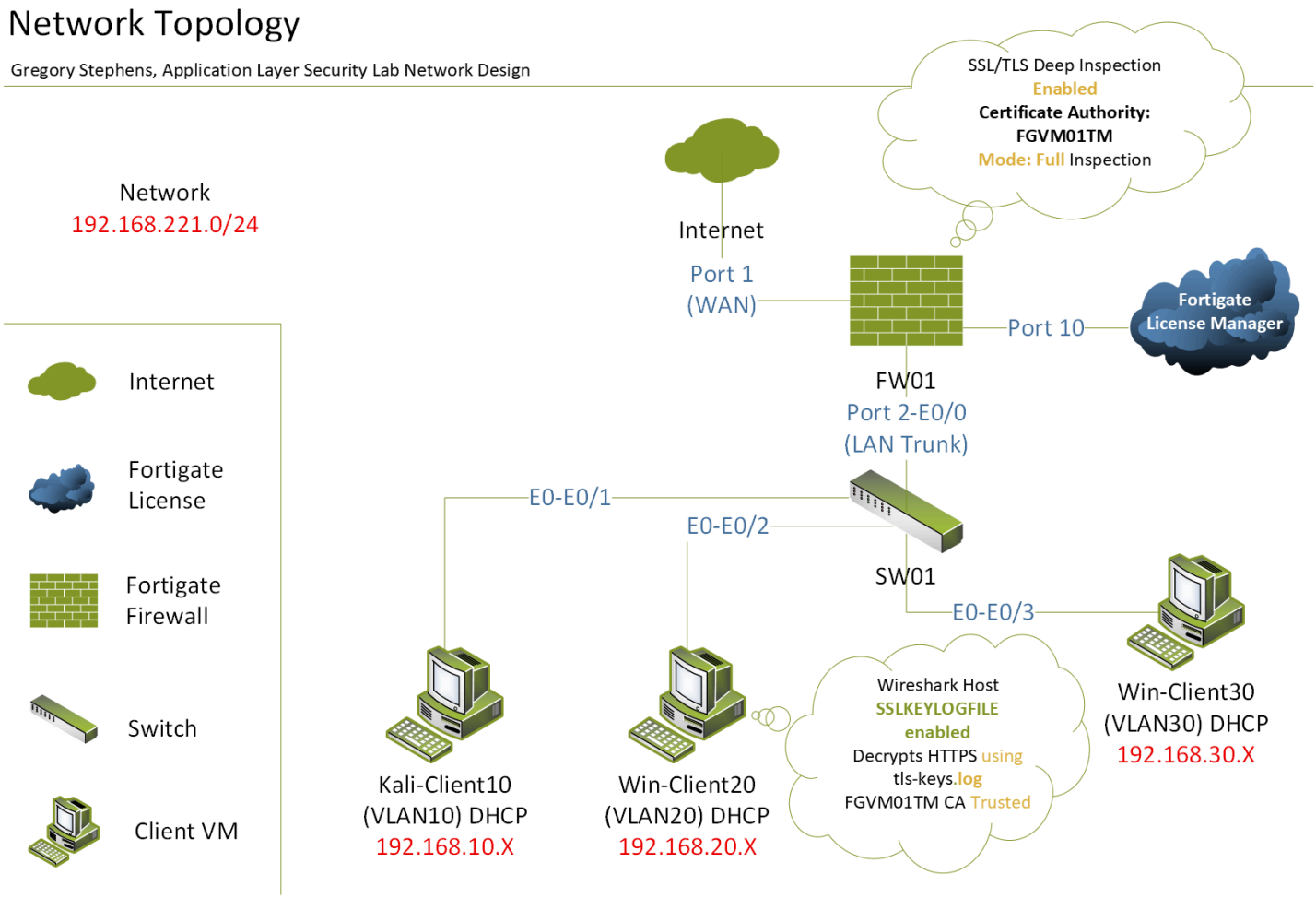
The virtual network was designed as follows:

- **FortiGate Firewall (FW01)** connected to:
    - **Port1 (WAN):** 192.168.221.133 – provides Internet access through the 192.168.221.0/24 network.
    - **Port2 (LAN Trunk):** connected to **SW01**, carrying VLANs 10, 20, and 30.
    - **Port10:** connected to the **FortiGate License Manager** for management and activation services.
- **Switch (SW01)** distributes traffic to three VLANs:
    - **VLAN10 – Kali-Client10:** DHCP-assigned address in 192.168.10.x, Gateway 192.168.10.1
    - **VLAN20 – Win-Client20 (Wireshark Host):** DHCP-assigned address in 192.168.20.x, Gateway 192.168.20.1
    - **VLAN30 – Win-Client30:** DHCP-assigned address in 192.168.30.x, Gateway 192.168.30.1

The FortiGate firewall was configured for **Full SSL Inspection** using its built-in CA certificate (`FGVM01TM…`) to re-sign intercepted HTTPS sessions. The CA was exported and trusted on the Windows host in VLAN20 to prevent browser certificate warnings.

Wireshark was run on the VLAN20 Windows client to capture traffic before and after decryption. A session key log file (`tls-keys.log`) was generated via the environment variable `SSLKEYLOGFILE` and imported into Wireshark, allowing inspection of decrypted HTTPS packets.

This setup successfully demonstrates the flow of encrypted traffic from client to Internet, decrypted and inspected by the FortiGate, and re-encrypted before reaching its destination, all within a secure, segmented virtual network.



Network Diagram

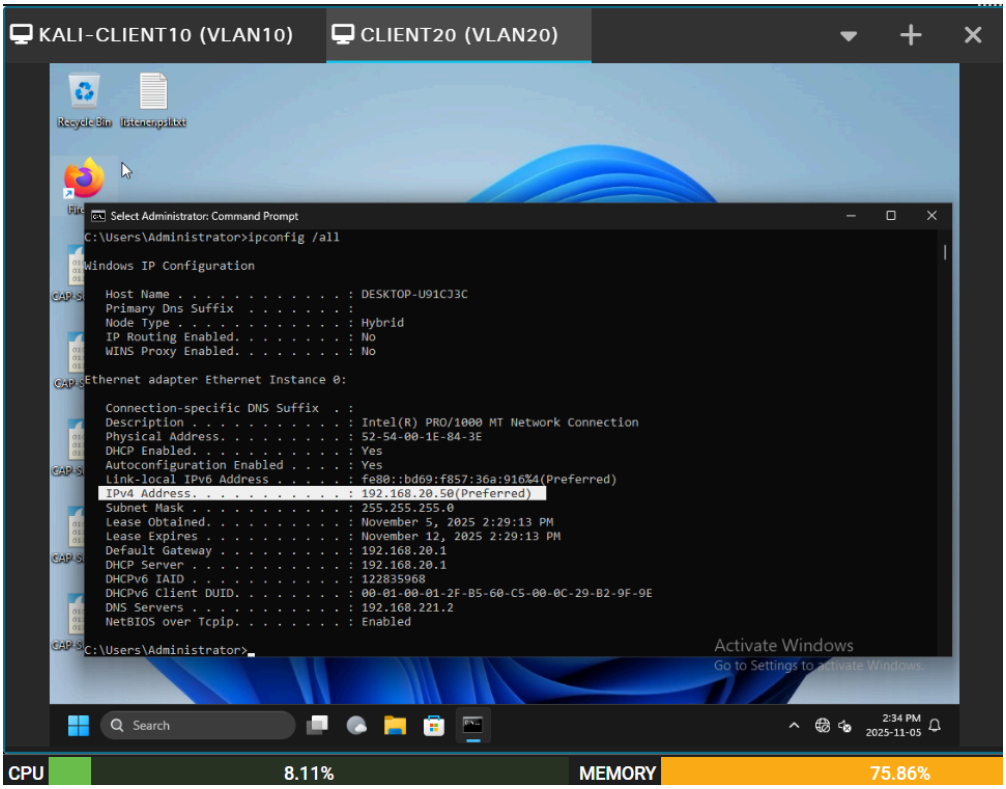
Figure 1 — Network topology showing VLAN segmentation and SSL/TLS deep inspection flow.

Step-by-Step Configuration

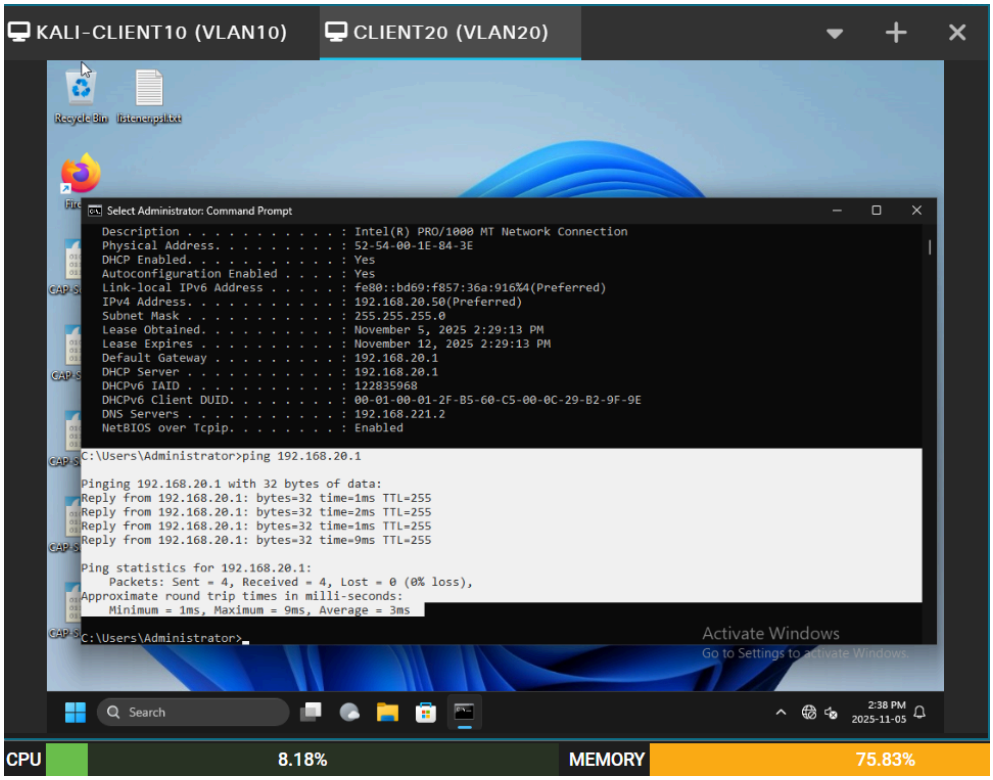
1 – Verify Network Connectivity

- **`ipconfig /all`** on Windows VLAN20 → confirm 192.168.20.50

- **`ping 192.168.20.1`** success



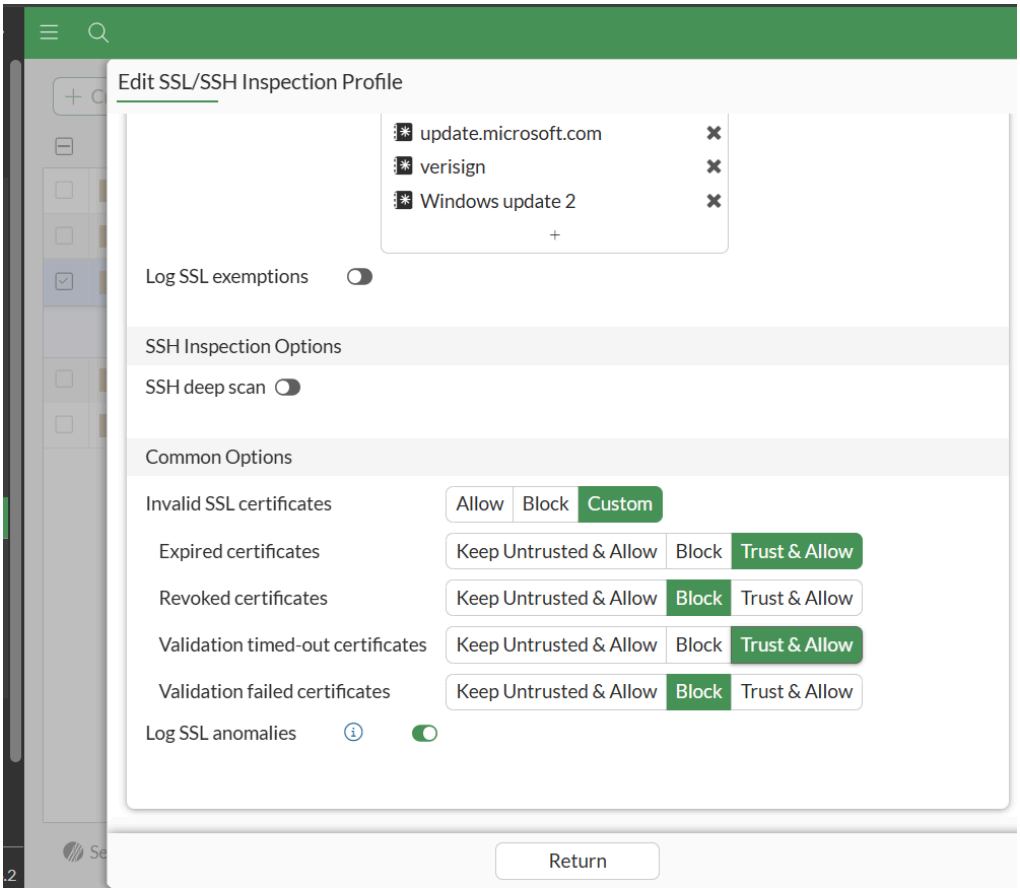
ss02_ipconfig_vlan20



ss03_ping_gw_vlan20

2 – Create SSL Deep Inspection Profile

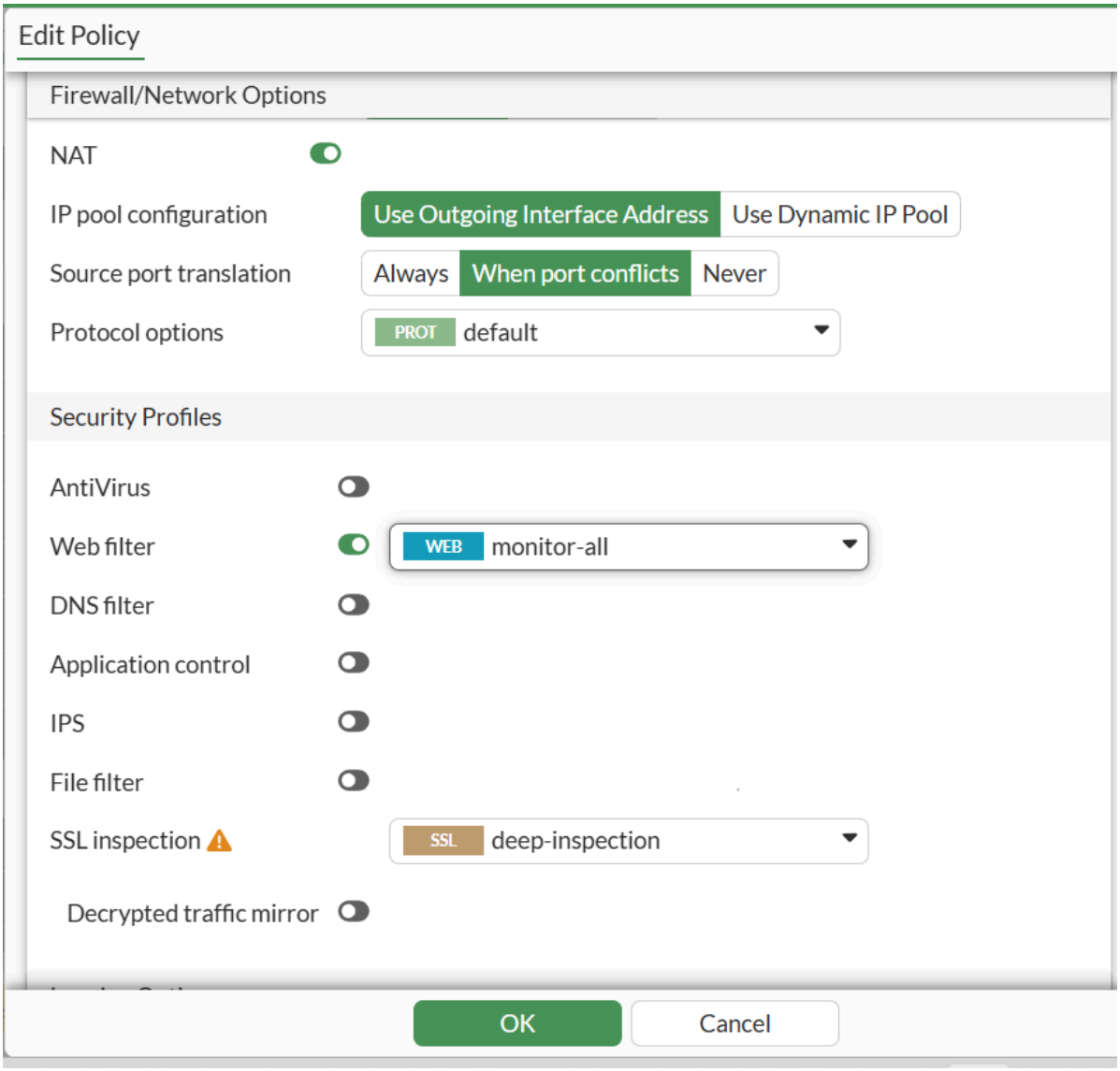
- Security Profiles → SSL/SSH Inspection → Clone "**deep-inspection**" as **deep-lab**
- Inspection Mode: **Full SSL Inspection**
- Invalid/Expired/Validation Failed: **Allow**
- Revoked: **Block**
- Validation Timeout: **Keep untrusted and allow**
- SNI Check: **Enable**
- CA Certificate: **FortiGate CA FGVM01TM…**



ss05_fgt_ssl_profile

3 – Apply Profile to Policy

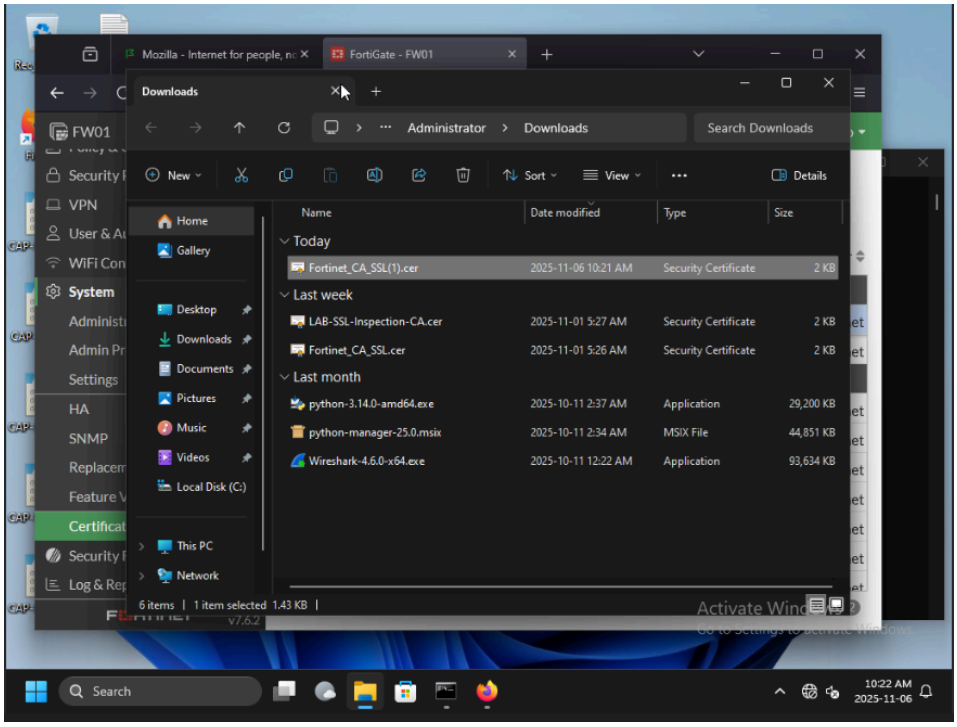
- Policy & Objects → IPv4 Policy
  - Source: **port2.20 Destination: port1 (WAN)**
  - NAT: **Enabled**
  - SSL/SSH Inspection: **deep-lab**
  - Web Filter: **None (or Monitor All → Allow on rating error)**



ss05_fgt_ssl_profile

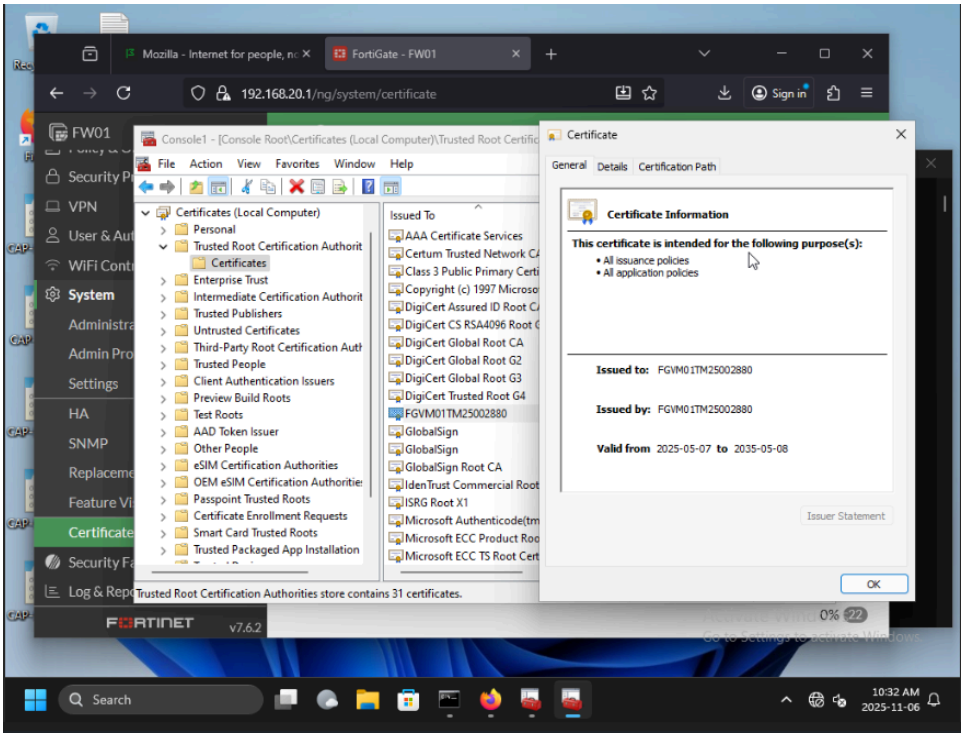
4 – Export and Trust FortiGate CA

- Download CA from **`https://192.168.20.1`**



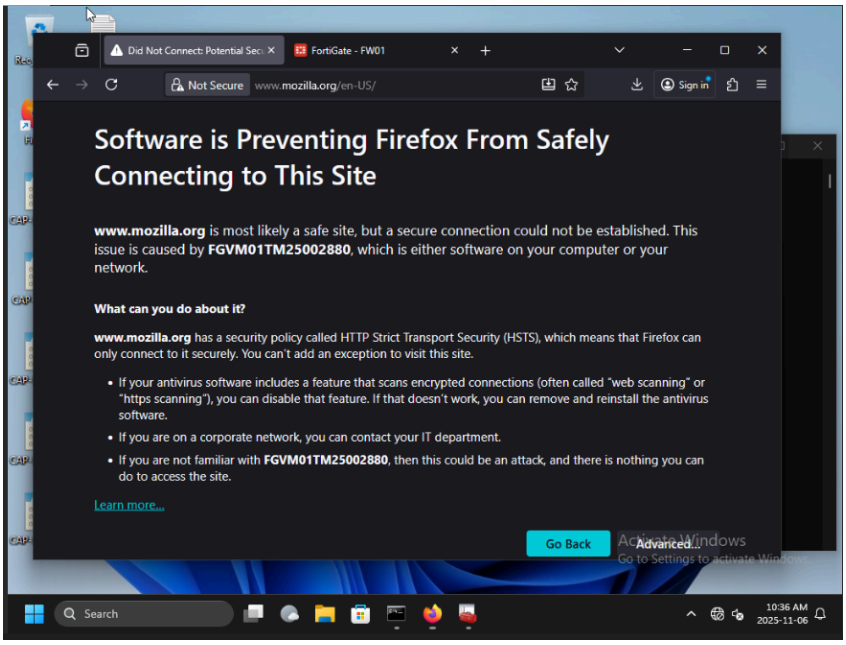
ss07_download.ca.from.fgt

- Import into Trusted Root Certification Authorities (MMC)



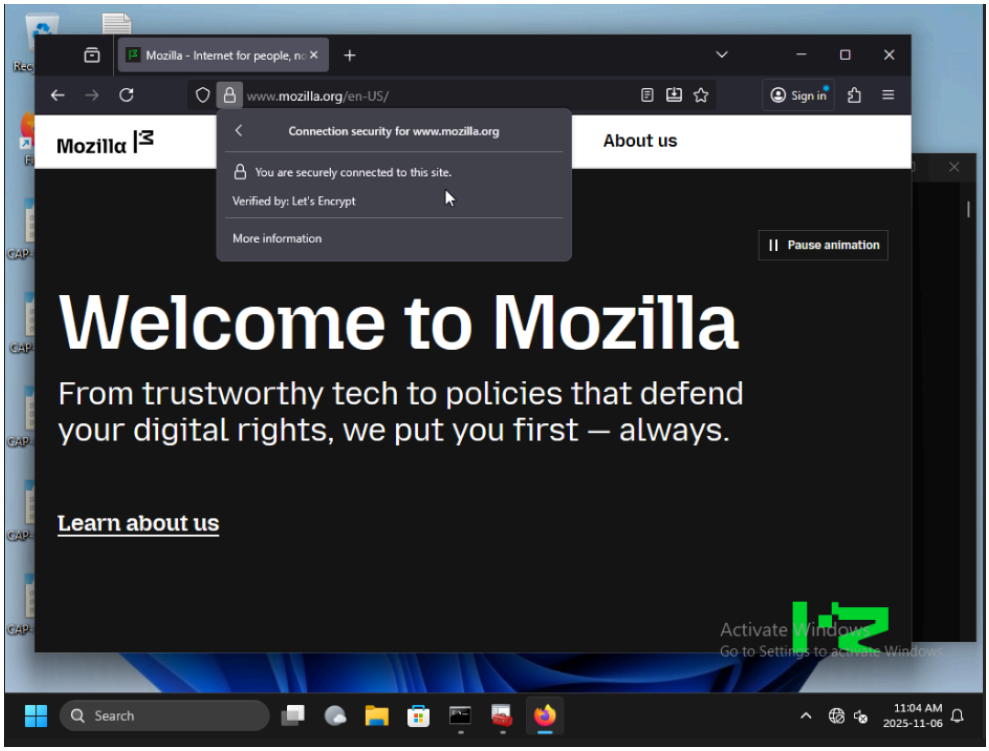
ss08_win_trusted_root

- Before CA trust → browser warning (https://www.mozilla.org)



ss04_internet_no_cert

- After CA trust → secure connection shown

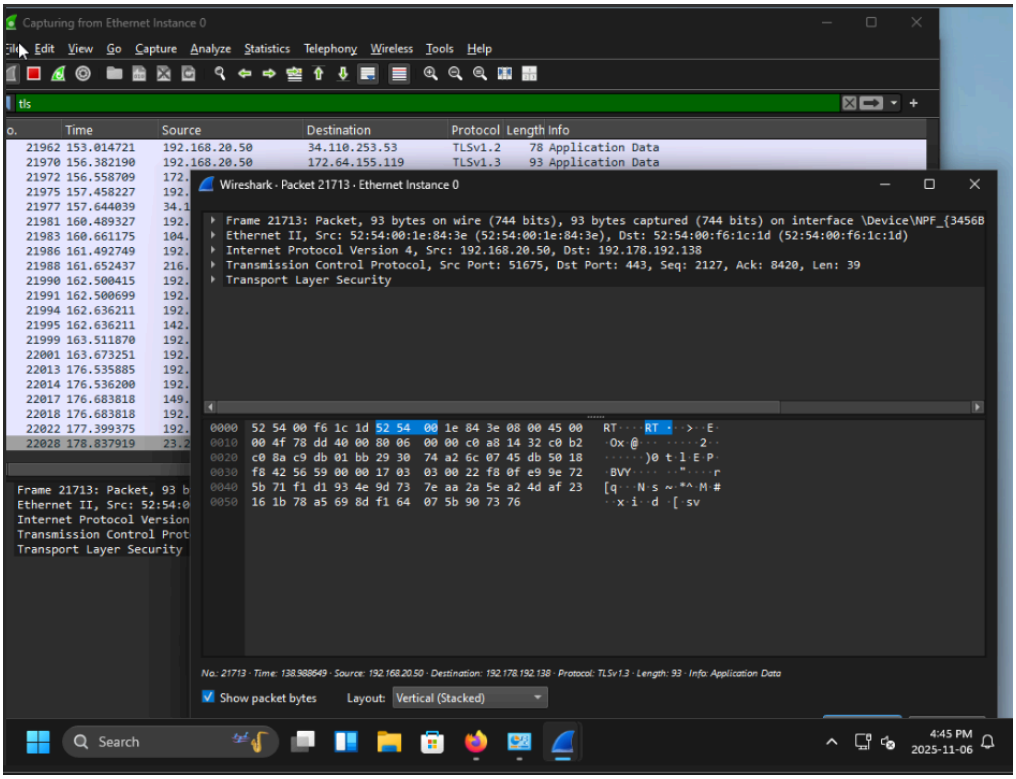


ss09_browser_secure_after_ca

5 – Wireshark TLS Decryption

A. Before Decryption

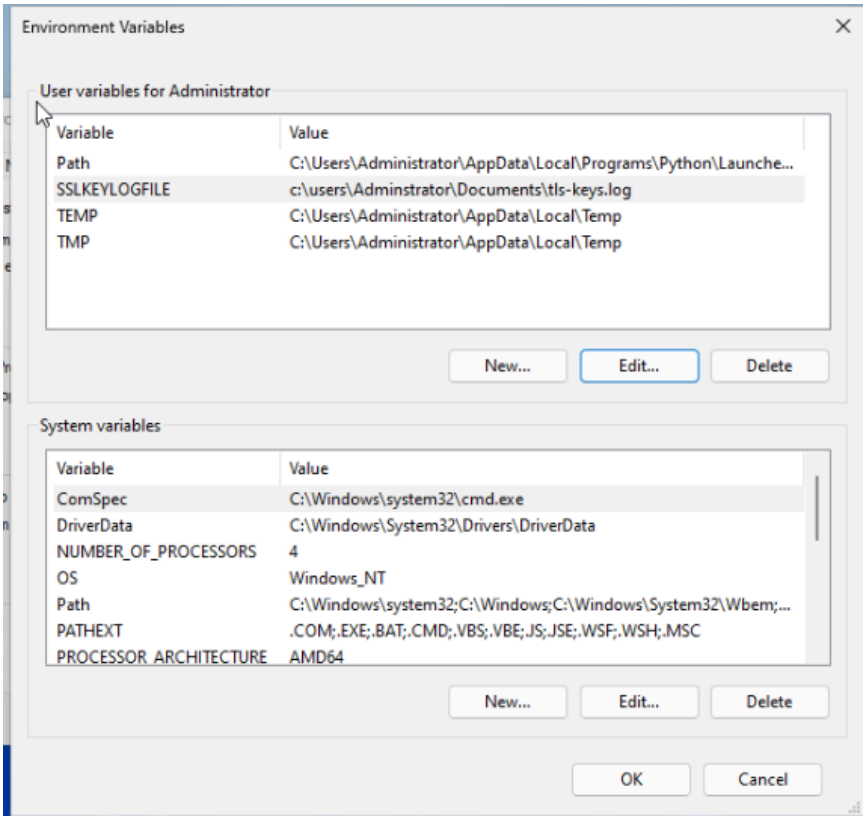
- Capture on Windows **VLAN20 NIC (192.168.20.50)**
- Filter **`tls`** → shows “Application Data” only



ss10_wireshark_before

B. Enable Key Logging

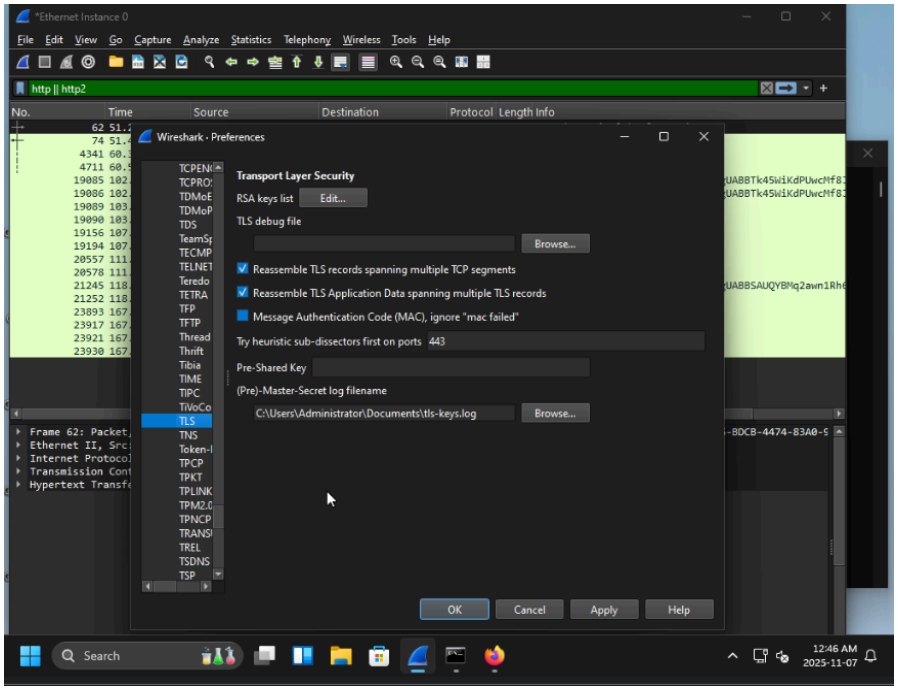
- Create Environment Variable **`SSLKEYLOGFILE = C:\Users\Administrator\Documents\tls-keys.log`**



ss11_st_sslkeylogfile

C. Configure Wireshark

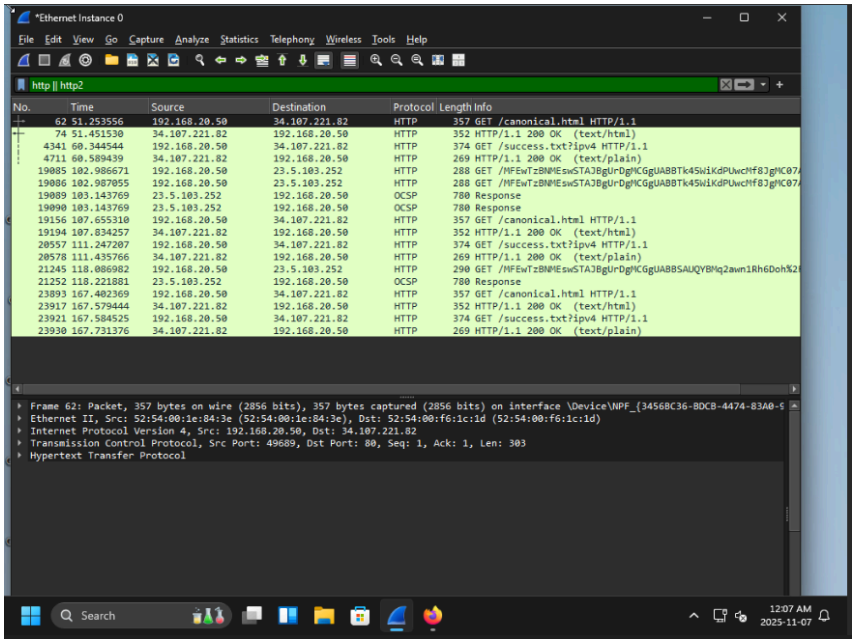
- Edit → Preferences → Protocols → TLS
- (Pre)-Master-Secret log filename → **`C:\Users\Administrator\Documents\tls-keys.log`**



ss12_wireshark_tls_key_log_pref

D. After Decryption

- Re-capture traffic, filter **`http || http2`**
- HTTP GET and 200 OK visible → proof of decryption



ss13_wireshark_after

Test Results Summary

Test	Expected Outcome	Result
HTTPS before cert	Browser warning	✅ Warning shown
HTTPS after cert	Trusted FortiGate CA, no warning	✅ Success
Wireshark (before)	Encrypted TLS Application Data only	✅ Confirmed
Wireshark (after)	Visible HTTP GET/response data	✅ Confirmed
SSL Inspection	Decrypt–Inspect–Re-encrypt functional	✅ Passed

Answers to Questions

1. Explain the process of SSL/TLS interception and include how certificates are handled and the role of root Certificate authorities in maintaining trust.

The firewall intercepts TLS by creating two encrypted sessions: client ↔ firewall and firewall ↔ server. It decrypts traffic, inspects for threats, and re-encrypts it. To keep browsers from warning, the firewall re-signs server certificates with its internal CA, which must be trusted on clients. Root CAs anchor global trust; in this lab, the FortiGate CA acted as the trusted root.

2. Technical challenges and solutions

Challenge	Impact	Solution / Workaround
Certificate pinning	Certain apps fail on re-signed certs	Bypass those domains or use endpoint agents
Encrypted SNI (TLS 1.3 ECH)	Hides the domain in the handshake	Use DNS logging or allowlists
QUIC / HTTP3 (UDP/443)	Skips TLS inspection	Disable QUIC or block UDP/443
Performance overhead	Decrypt/re-encrypt load	Use hardware offload or limit scope
FortiGuard offline (lab)	False UTM blocks	Allow On rating error in the profile

3. Privacy and compliance (GDPR, HIPAA, PIPEDA)

SSL inspection can expose private data, so its use must meet legal and ethical standards.

- **GDPR/PIPEDA:** allow only when necessary for network security; provide user notice and bypass sensitive categories (banking, medical).

- **HIPAA:** decrypting PHI requires controls and auditing; many organizations exclude patient portals entirely.
  Proper documentation and policy limits ensure compliance and protect user privacy.

Conclusion

Full SSL/TLS deep inspection was successfully implemented on a FortiGate firewall in CML. After trusting the FortiGate CA, browsers established secure sessions, and Wireshark confirmed decrypted HTTPS payloads using exported TLS keys.

The lab validated how firewalls securely inspect encrypted traffic while emphasizing the privacy and compliance constraints that accompany this capability.

References

- Fortinet. (2025). *FortiGate SSL and SSH Inspection Guide*. Retrieved from https://docs.fortinet.com/document/fortigate