


TLENVOY - A Tool for TLS Termination and Inspection

Udit Misra

Department of Computer Science, North Carolina State University

Article Info	ABSTRACT
<p>Article history:</p> <p>Received Mar, 2025 Revised Apr, 2025 Accepted Apr, 2025</p> <hr/> <p>Keywords:</p> <p>Envoy; Forward Proxy; Network Security; Networking; Packet Analysis; TLS</p>	<p>With the increasing adaption on encrypted communication over the internet, ensuring the security over network traffic has become very crucial. Transport Layer Security (TLS) is now widely used to secure data in transit, but at the same time it also poses challenges for network administrators who need to inspect traffic for malicious content or policy violations. This paper explores the use of Envoy, an open-source edge and service proxy, as a forward proxy to inspect TLS traffic. By leveraging Envoy's capabilities, organizations can maintain a secure environment for all nodes behind the proxy. We discuss the architecture, implementation, security considerations, and potential challenges of using Envoy for TLS inspection. The paper concludes with recommendations for deploying such a system in a secure and efficient manner.</p> <p><i>This is an open access article under the CC BY-SA license.</i></p> <div></div>
<p>Corresponding Author:</p> <p>Name: Udit Misra Institution: Department of Computer Science, North Carolina State University Email: umisra@alumni.ncsu.edu</p>	

1. INTRODUCTION

The wide use of encrypted communication protocols, particularly TLS [1], has significantly enhanced the privacy and security of internet traffic [2], [3]. However, this encryption also complicates the task of monitoring and securing network traffic within organizational boundaries. Malicious actors can easily exploit the encrypted channels to exfiltrate data or deliver malware within the secure networks, making it essential for organizations to inspect TLS traffic without compromising security.

A forward proxy acts as an intermediary between internal clients and external servers, allowing organizations to control and monitor outbound traffic. By intercepting and decrypting TLS traffic, a forward proxy can inspect the contents of

encrypted communications, ensuring that they comply with security policies. This paper examines the use of Envoy as a forward proxy for TLS inspection, focusing on its architecture, implementation, and the security implications of such a system.

Importance of TLS Inspection

TLS inspection is crucial in networking for enhancing security, compliance and visibility. With such a significant portion of web traffic being encrypted, cybercriminals often hide malware, phishing attacks, and other threats within the encrypted data. TLS inspection allows organizations to decrypt, scan, and re-encrypt traffic, ensuring that security tools like firewalls and intrusion detection systems can effectively identify and block malicious content. It also provides critical visibility into

encrypted traffic, helping organizations enforce security policies and prevent data breaches. Compliance with industry regulations such as GDPR, HIPAA, and PCI-DSS is another key benefit, as TLS inspection aids in monitoring sensitive data to prevent unauthorized access and data loss [4]. Additionally, it helps mitigate man-in-the-middle attacks, ensures proper content filtering, and supports advanced security architectures like Zero Trust by continuously verifying internal and external traffic. However, implementing TLS inspection comes with challenges, including performance overhead due to decryption and re-encryption, privacy concerns regarding sensitive data, and the complexity of certificate management. Despite these challenges, TLS inspection remains a vital security measure, offering organizations enhanced threat detection, improved compliance, and better control over encrypted traffic while maintaining a balance between security, performance, and privacy.

2. BACKGROUND

2.1 Forward Proxy

A forward proxy is a server that sits between internal clients and external servers, acting as an intermediary for outbound requests. It can be used to enforce security policies, filter content, and monitor traffic. When configured to inspect TLS traffic, the forward proxy decrypts the traffic, inspects it, and then re-encrypts it before forwarding it to the destination server.

2.2 Envoy

Envoy is an open-source edge and service proxy designed for cloud-native applications. It provides a high-performance, extensible platform for managing traffic, including load balancing, observability, and security features. Envoy's support for modern protocols, including HTTP/2, gRPC, and TLS, makes it an ideal choice for implementing a forward proxy with TLS inspection capabilities.

2.3 TLS

TLS is a cryptographic protocol suite designed to provide confidentiality and data integrity between two communicating entities. When properly implemented, TLS prevents any third party from accessing the application layer payload, even if they intercept the traffic [5]. However, there are scenarios where network administrators need to inspect encrypted traffic, such as preventing malware from entering an organization's network. To address this need, TLS interception proxies were developed, which essentially perform a controlled man-in-the-middle (MitM) operation.

In this approach, network operators configure a proxy that intercepts and decrypts TLS traffic in a way that is transparent to users, provided they have consented to it. This is typically done by having clients install a trusted Certificate Authority (CA) certificate issued by the proxy [6]. Once in place, all network traffic is routed through this interception proxy, which establishes separate TLS connections with both the client and the destination server. As a result, the proxy is able to decrypt, inspect, and then re-encrypt the data before forwarding it. To the client, the proxy appears as the destination server since it dynamically generates certificates for requested resources. The client, having already trusted the proxy's CA certificate, accepts these certificates without issue. Essentially, the proxy acts as both a TLS client (when communicating with the web server) and a TLS server (when communicating with the client), allowing traffic to be read in plaintext before being relayed.

Despite its usefulness, TLS interception introduces several security and trust challenges. It disrupts the client's assumptions about the security of its connection, as the proxy effectively breaks end-to-end encryption. This means the client can no longer be certain of the cryptographic protocols in use, as the

proxy might downgrade security by utilizing weaker encryption methods or accepting compromised certificates from the server. To maintain trust and security, network operators must ensure that the interception proxy upholds the expected cryptographic standards and does not introduce vulnerabilities.

3. ARCHITECTURE OF TLS INSPECTION USING ENVOY AS A FORWARD PROXY

3.1 Overview



Figure 1: TLS connection with an Interceptor Proxy

The diagram illustrating TLS inspection using Envoy as a forward proxy, as shown in Figure 1, consists of several key components, each playing a

crucial role in handling and securing outbound traffic. Figure 2 provides a detailed breakdown of these components and their functions [7].

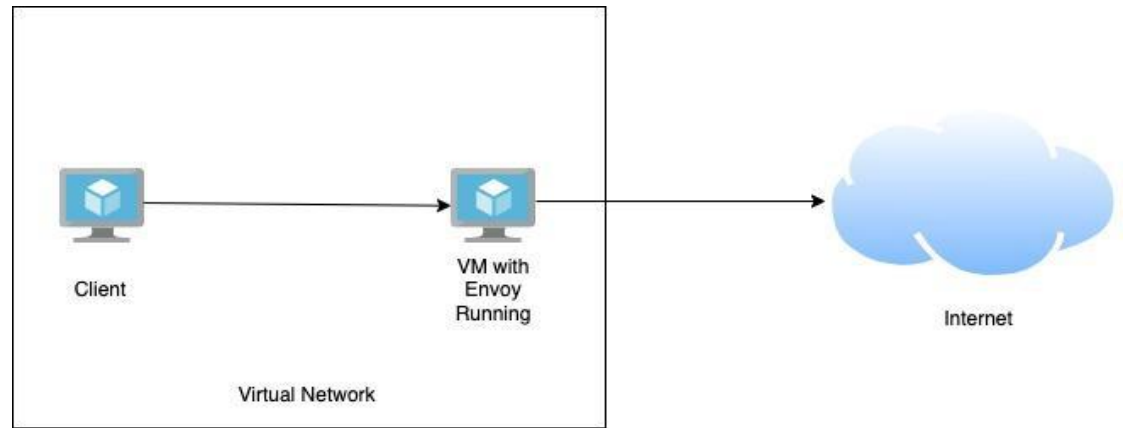


Figure 2: Architecture Overview Showing Traffic Flow

Client: The client is the internal node that initiates outbound HTTPS requests. In our setup, we used a Virtual Machine (VM) hosted in a public cloud environment to generate these requests. The VM serves as a representative internal client that requires internet access, simulating real-world enterprise environments where users or applications need to communicate securely with external services.

Envoy Forward Proxy: Acting as an intermediary, the Envoy forward proxy intercepts, decrypts, inspects, and

then re-encrypts the network traffic before forwarding it to its intended destination. In our implementation, Envoy was deployed on a Linux machine within the same virtual network as the client. This placement ensures that all outbound requests from the client are first routed through the Envoy proxy, allowing for deep packet inspection and enforcement of security policies.

Certificate Authority (CA): The Certificate Authority is responsible for issuing digital certificates used by the forward proxy to impersonate

destination servers during TLS interception. In our setup, we created a self-signed certificate to facilitate secure communication between the client and the proxy. However, in real-world deployments, enterprises typically use certificates issued by a trusted CA to ensure authenticity, security, and compliance with organizational and industry standards.

Security Policies: These refer to the network rules and configurations that dictate how traffic should be inspected, controlled, and forwarded. For our implementation, we enforced a routing policy where all outbound traffic from the client was explicitly directed to the Envoy proxy. This was achieved by deploying Envoy within the same virtual network as the client, ensuring that no direct internet access was possible. Once intercepted, the proxy inspects the encrypted traffic before forwarding it securely to external destinations over the internet.

3.2 Workflow

- 1) **Client Request:** The client initiates a request to an external server.
- 2) **Interception:** Envoy intercepts the request and establishes a TLS connection with the client, using a self signed certificate by client
- 3) **Decryption:** Envoy decrypts the traffic using the private key corresponding to the certificate.
- 4) **Inspection:** The decrypted traffic is inspected for malicious content, policy violations, or other anomalies.
- 5) **Re-encryption:** Envoy establishes a new TLS connection with the destination server and re-encrypts the traffic.
- 6) **Forwarding:** The inspected traffic is forwarded to the destination server.
- 7) **Response Handling:** The response from the destination server follows the same process in reverse, with Envoy decrypting, inspecting, and re-encrypting the response before sending it back to the client.

4. RESULTS

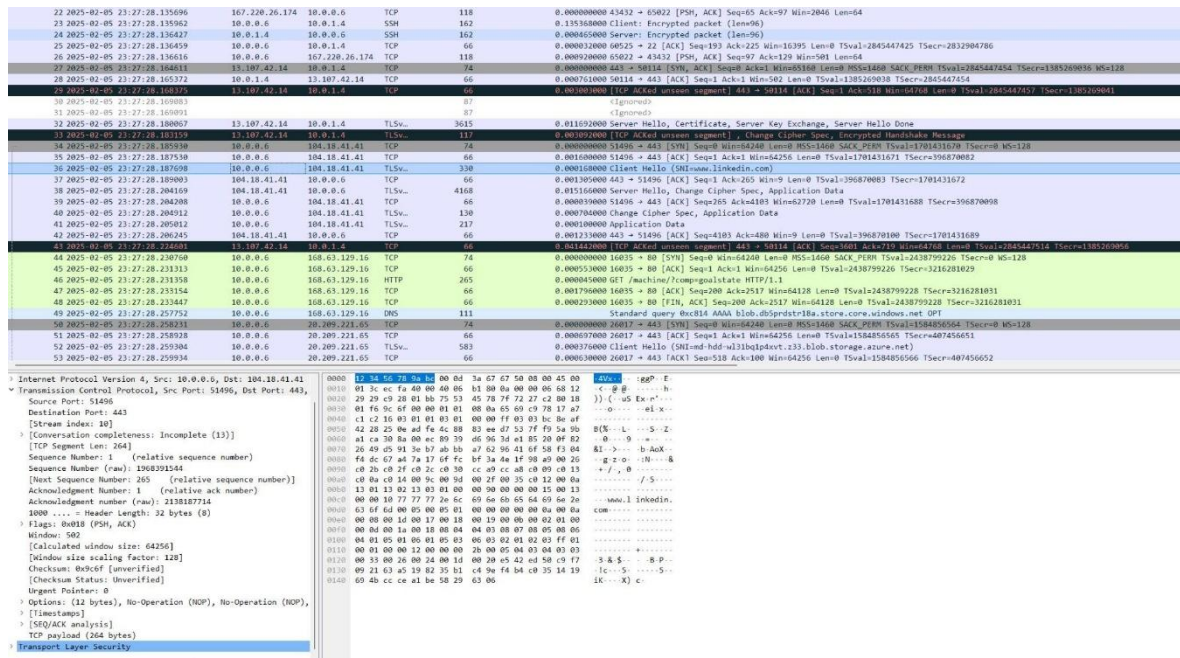


Figure 3: Wireshark packet capture at Envoy

Figure 3 displays the packet capture output obtained from the Envoy proxy server, showcasing the TLS inspection process. The

capture reveals the termination of encrypted traffic from the client, followed by the establishment of a new TLS connection

between the proxy and the destination server on the internet.

In this setup, the client, with an internal IP address of 10.0.1.4, forwards its traffic to the Envoy proxy, which is assigned the IP 10.0.0.6. The Envoy proxy then initiates a new TLS connection to the external server with the public IP 104.18.41.41. The packet capture provides clear evidence of two distinct TLS handshakes—one occurring between the client and the Envoy proxy, and another between the proxy and the destination server—confirming the interception and inspection of encrypted traffic before it is securely forwarded.

5. CONCLUSION

Inspecting TLS traffic using Envoy as a forward proxy is a powerful tool for maintaining a secure environment within an organization. By decrypting, inspecting, and re-encrypting traffic, organizations can detect and prevent malicious activity while enforcing security policies. However, the implementation of such a system requires

careful consideration of certificate management, performance impact, security policies, and privacy concerns.

To successfully deploy Envoy for TLS inspection, organizations must address the challenges and limitations associated with this approach, including the need to protect the proxy from attacks, ensure data integrity, and comply with legal and ethical standards. With proper planning and implementation, Envoy can provide a secure and efficient solution for inspecting TLS traffic in a modern network environment.

6. FUTURE WORK

Future research could explore the integration of machine learning and artificial intelligence techniques to enhance the detection of malicious content within inspected traffic. Additionally, the development of more efficient cryptographic methods for TLS inspection could help reduce the performance overhead associated with decryption and re-encryption.

REFERENCES

- [1] E. Rescorla, "The transport layer security (TLS) protocol version 1.3," 2018.
- [2] S. Dyllan, H. Dahimene, P. Wright, and P. Xiao, "Analysis of HTTP and HTTPS usage on the university internet backbone links," *J. Ind. Intell. Inf. Vol.*, vol. 2, no. 1, 2014.
- [3] D. Naylor *et al.*, "The cost of the 's' in https," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 133–140.
- [4] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," 2008.
- [5] R. Fielding *et al.*, "Hypertext transfer protocol--HTTP/1.1," 1999.
- [6] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile," 2008.
- [7] H. Krawczyk and P. Eronen, "HMAC-based extract-and-expand key derivation function (HKDF)," 2010.