International Journal Software Engineering and Computer Science (IJSECS)
5 (2), 2025, 887-896
Published Online August 2025 in IJSECS (http://www.journal.lembagakita.org/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v5i2.5153.

RESEARCH ARTICLE Open Access

Optimization of Data Security Protection with Full SSL Inspection on AWS Using FortiGate Virtual Appliance

Yuma Akbar
Information Systems Study Program, Faculty of Computer Science, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.
Email: yuma.pjj@gmail.com.

Gipari Pradina Abdillah *
Information Systems Study Program, Faculty of Computer Science, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.
Corresponding Email: giparipradinaabdillah@yahoo.com.

Received: July 14, 2025; Accepted: July 25, 2025; Published: August 1, 2025.

Abstract: The expanding adoption of cloud services, particularly Amazon Web Services (AWS), has intensified challenges in protecting encrypted data traffic within network security frameworks. SSL/TLS protocols, widely utilized for data encryption, have become exploitation vectors for cyber adversaries as conventional security solutions lack the capability to scrutinize encrypted traffic effectively. The research addresses such security gaps by implementing Full SSL Inspection through Fortigate Virtual Appliance deployment within AWS cloud environments. The study examines cloud-based network architecture integrated with Fortigate systems, employing methodologies that encompass virtual appliance installation, SSL/TLS inspection feature configuration, and assessment of system effectiveness alongside performance impact evaluation. Research instruments include simulated cyber-attack scenarios targeting encrypted traffic patterns. Findings demonstrate that Full SSL Inspection significantly enhances threat detection capabilities within network traffic, albeit with measurable increases in system latency and computational overhead. The implementation of Fortigate Virtual Appliance proves effective in strengthening AWS data security postures. Research outcomes emphasize the necessity for configuration optimization to maintain security-performance equilibrium, positioning the solution as viable for organizations prioritizing data protection strategies.

Keywords: SSL Inspection; Fortigate 7.6; AWS; Encryption; Decryption; Cloud Security.

1. Introduction

The rapid advancement of information and communication technology has significantly escalated the need for data security within network environments. Organizations and enterprises worldwide increasingly rely on cloud computing services to store and manage their critical data. Amazon Web Services (AWS), as one of the leading cloud computing service providers, offers adequate infrastructure-based solutions for various modern business requirements [1][8]. However, with the growing complexity of cyber attacks against network systems, optimal data security has become one of the greatest challenges faced in managing information within cloud computing environments. The evolution of cyber threats has driven the necessity for more sophisticated security approaches, particularly in confronting attacks that exploit encryption to conceal malicious activities. One crucial technique in network data security is SSL Inspection, which enables analysis and monitoring of traffic encrypted using SSL/TLS protocols. SSL Inspection allows network administrators to detect and address potential threats hidden within encrypted traffic, which is increasingly utilized by modern web applications to protect data communication [3][4][5].

SSL/TLS protocols, while designed to provide communication security, can be exploited by cybercriminals to hide malicious payloads from conventional security system detection. Such phenomena create "blind spots" in organizational security infrastructure, where encrypted traffic can bypass firewalls and intrusion detection systems without adequate inspection. Therefore, SSL Inspection implementation becomes crucial for maintaining full visibility across all network traffic. Nevertheless, full SSL Inspection deployment requires careful management, particularly regarding system performance and scalability that may affect network operational efficiency. The SSL/TLS traffic decryption and re-encryption process demands significant computational resources, which can impact network latency and throughput if not properly configured.

On the other hand, utilizing appropriate cloud security tools such as Fortigate Virtual Appliance from Fortinet, which can operate on AWS, becomes an effective solution for mitigating various network threats (Fortinet, 2025) [6]. Fortigate Virtual Appliance represents a firewall and network security solution that offers robust protection, including SSL Inspection management, to ensure data residing in cloud environments remains secure from threats such as malware, phishing, and other attacks. The solution integrates various security features within a single platform, including intrusion prevention system (IPS), antivirus, web filtering, and application control. Fortigate Virtual Appliance advantages lie in its capability to deliver enterprise-grade security within flexible and scalable cloud environments. The platform supports rapid deployment and configuration that can be customized to specific organizational needs, while maintaining security policy consistency across hybrid cloud environments.

However, despite Fortigate Virtual Appliance's advanced features, challenges emerge in data protection optimization processes, particularly regarding SSL Inspection management efficiency [1][2]. Many organizations and enterprises experience difficulties implementing security systems that can manage SSL/TLS traffic efficiently without sacrificing system performance or adding unwanted latency. Configuration complexity, certificate authority management, and inspection policy fine-tuning become factors requiring specialized expertise. Additional challenges arise from the need to balance high security levels with optimal user experience. Improper SSL Inspection implementation can cause application performance degradation, compatibility issues with certain applications, and even privacy concerns related to sensitive traffic decryption. Within such framework, the research focuses on data security protection optimization using Full SSL Inspection in AWS Cloud by leveraging Fortigate Virtual Appliance. The study aims to analyze how simple configuration using three virtual machines (VMs) in AWS Cloud—the first VM as Fortigate Virtual Appliance, the second VM as Ubuntu Desktop-based testing host, and the third VM as simple web server—can optimize SSL Inspection processes in enhancing data protection [6].

The research approach adopts practical methodology that can be implemented by organizations with various infrastructure complexity levels. By using relatively simple yet representative architecture, the study will explore various technical aspects of SSL Inspection implementation, including certificate management, policy configuration, performance tuning, and security effectiveness measurement. Using such structured approach, the research will evaluate the extent to which SSL traffic management can be effectively conducted on AWS, utilizing efficient security devices without affecting overall system performance. Evaluation will encompass aspects such as detection rates against various threat types, network latency impact, resource utilization, and scalability considerations. The solution is expected to provide practical guidance beneficial for organizations and enterprises using AWS Cloud in strengthening data security protection efforts, while serving as reference for security implementation in broader cloud environments. Research findings are also expected to contribute to best practices development for SSL Inspection implementation in cloud environments, particularly within AWS infrastructure.

2. Related Work

The academic literature surrounding SSL/TLS inspection and cloud security implementations has evolved substantially over the past decade, establishing multiple research streams that inform current security practices. Foundational research in SSL/TLS security has established critical principles for secure communication protocols. Amir Ibrahim (2020) examined Secure Socket Layer fundamentals and certificate verification processes, establishing that proper SSL implementation serves as a primary defense mechanism against network-based attacks [9]. The research demonstrated how certificate validation procedures directly impact communication security effectiveness. Yu Dun-Yi (2020) advanced the field by developing data encryption methods for SSL digital authentication signature systems, focusing on privacy protection principles [18]. The study addressed the growing tension between security requirements and user privacy expectations in modern SSL implementations. Pavel Razumov *et al*. (2023) investigated web application security operating under SSL/TLS protocols, proposing enhanced security frameworks for web-based services [19]. Their work identified specific implementation vulnerabilities that attackers exploit to circumvent traditional security measures.

Research into SSL attack methodologies has revealed critical weaknesses in standard implementations. Nathanael Dharmawan *et al*. (2022) conducted security analysis of university networks through SSL/TLS attack simulations, identifying significant vulnerabilities in academic network infrastructures [16]. The study revealed that educational institutions frequently lack adequate SSL inspection capabilities, creating exposure to encrypted malicious traffic. Siromani Duddu *et al*. (2020) investigated SSL stripping attacks using ARP spoofing techniques, demonstrating how attackers exploit SSL implementation weaknesses [27]. Their research showed that even properly configured SSL systems remain vulnerable to sophisticated attack vectors. Vinod S. Khandkar and Manjesh K. Hanawal (2021) examined host identity masking through encrypted TLS/SSL handshakes, revealing how encryption technologies can serve dual purposes for both protection and concealment [24]. The study highlighted the paradoxical nature of encryption in modern security architectures.

Practical implementations of Fortigate technology have been documented across diverse organizational environments. Fauzi Rizki Arbie and Mugi Raharjo (2024) examined network security implementation using security profiles through Fortigate systems at government institutions [12]. Their findings indicated that proper configuration significantly enhances security layers while maintaining operational efficiency across enterprise networks. Gipari Pradina Abdillah *et al*. (2024) focused on VPN IPSec tunnel optimization using AES encryption through Fortigate devices, demonstrating measurable improvements in secure remote access capabilities [14]. The research showed quantifiable performance gains in encrypted tunnel implementations. Dhanu Prima Jaya *et al*. (2021) implemented computer network security using Fortigate as firewall systems in educational computer laboratories, documenting successful deployment methodologies in academic environments [21]. Their work provided practical guidance for educational institution security implementations.

Modern traffic analysis techniques have evolved to address sophisticated threat landscapes. Andi Dinda Nurul Fauziah *et al*. (2022) analyzed SD-WAN traffic steering technology implementation using FortiGate devices, demonstrating improved network performance and security through intelligent traffic management [17]. The research showed how advanced routing algorithms enhance both security and performance metrics. Jyoti Pandey *et al*. (2023) assessed deep packet inspection systems for network traffic and anomaly detection, providing insights into advanced traffic analysis capabilities [29]. Their study revealed that machine learning integration with traditional inspection methods significantly improves threat detection accuracy. Ergest Alite *et al*. (2020) developed deep SSL inspection systems with Active Directory integration, demonstrating seamless security implementation in Windows-based enterprise environments [28]. The research showed how SSL inspection can be effectively integrated with existing authentication infrastructures.

Cloud security research has addressed specific challenges in AWS environments. Neha Kewate *et al*. (2022) reviewed AWS cloud computing technology, identifying security challenges that affect optimal data protection deployment [11]. Their analysis revealed key areas where traditional security approaches require adaptation for cloud environments. Iqra Naseer (2023) examined AWS Cloud Computing Solutions optimization for business implementations, demonstrating how proper configuration enhances security implementations for organizations leveraging cloud computing [13]. The research established that AWS-Fortigate integration provides viable solutions for effective cloud security management. Muhammad Syahrul Mubarok and Muhammad Izman Herdiansyah (2023) implemented AWS cloud computing for hotel room reservation web applications, providing practical cloud deployment scenarios [20]. Their work demonstrated real-world AWS implementation challenges and solutions.

Recent research has explored artificial intelligence integration in cloud security systems. Venkata Ramana Gudelli (2023) introduced AI-powered insights for performance optimization in AWS cloud environments, showing how machine learning enhances workload management and affects performance in cloud settings [10]. The study demonstrated that intelligent resource allocation directly impacts SSL inspection efficiency and overall network performance. Balajee R M and Jayanthi Kannan M K (2023) developed intrusion detection

systems for AWS Cloud using hybrid deep learning algorithms, achieving improved detection rates for sophisticated attacks [23]. Their research showed that machine learning integration with traditional firewall systems enhances threat identification capabilities.

Industry documentation has provided practical guidance for SSL inspection implementation. Trend Micro (2017) published threat protection system SSL inspection best practices, offering practical guidance for implementing SSL inspection in enterprise environments [30]. The documentation addressed common implementation challenges and provided solutions for maintaining security without compromising performance. Ali Al-Mohamad (2024) conducted performance evaluation of firewall technologies, providing benchmarks for different firewall implementations and their effectiveness in various network environments [26]. The research offered valuable insights into selecting appropriate firewall technologies based on specific organizational requirements.

Specialized implementations have addressed specific organizational needs. Sari Dewi and Adam Iqbal Islami (2021) implemented web filtering using FortiGate FG300D routers, demonstrating effective content filtering capabilities [22]. Dadang Iskandar Mulyana *et al.* (2024) explored network security optimization for WiFi environments, implementing DNS filtering mechanisms to block malicious content using Orange Pi devices [15]. Hari Suryantoro *et al.* (2021) applied Fortigate technology in building IPSec-based VPN-IP networks, showing successful secure remote access implementations [25]. Their work provided practical methodologies for VPN deployment in enterprise environments. Alfry Aristo Jansen Sinlae *et al.* (2024) created computer network textbooks that provide theoretical foundations and practical guidance for network security implementation [31], serving as valuable resources for both academic and professional development in network security fields.

While existing literature covers various aspects of SSL inspection and cloud security, several gaps remain unaddressed. Current research lacks specific focus on optimizing Full SSL Inspection implementation for AWS environments using Fortigate Virtual Appliances. Most studies examine either SSL inspection or cloud security independently, without addressing their integration challenges. Performance optimization studies typically focus on either network security or cloud computing performance, but rarely examine the intersection of both domains. Additionally, practical implementation guides for small-to-medium scale deployments remain limited, with most research focusing on large enterprise implementations. The current research addresses these gaps by examining Full SSL Inspection optimization specifically within AWS Cloud environments using Fortigate Virtual Appliance, providing practical implementation guidance for organizations seeking to enhance their cloud security posture through effective SSL traffic management.

3. Research Method

This study employs the SPDLC (Security Policy Development Life Cycle) methodology, which proves suitable for presenting development stages of systems related to network security aligned with the research focus. SPDLC establishes strategies for organizational updates of network systems, with the network system development cycle defined across multiple phases. According to Luay A. Wahsheh and Jim Alves Foss (2008) cited from the thesis (Lukman & Suci, 2020) [31][32]. Goldman and Rawles 2024 describe SPDLC as stages beginning with evaluation phases that validate effectiveness from initial analysis stages [33]. Feedback from evaluation can impact changes in current architecture and technology implementations.

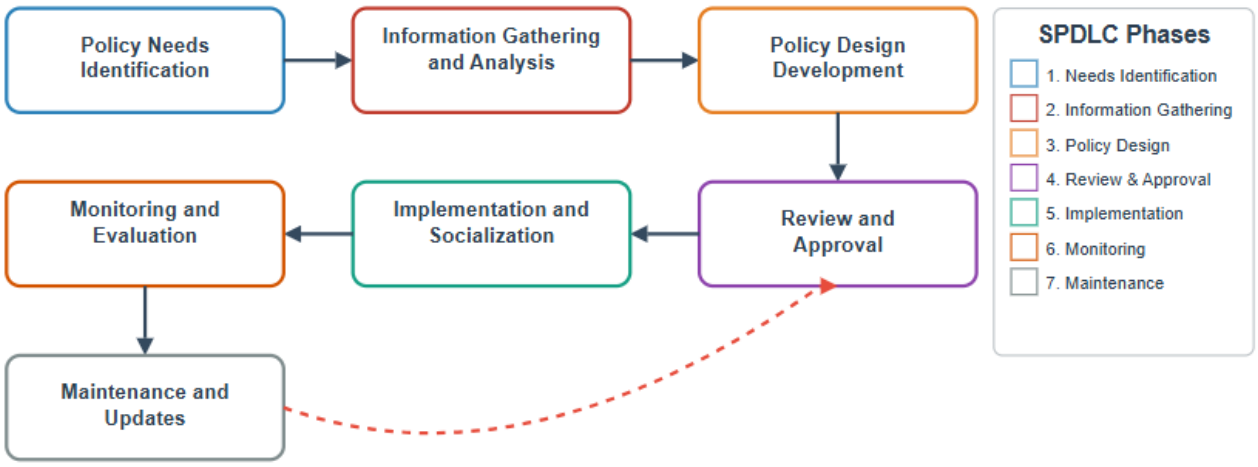

Figure 1. SPDLC Method Diagram

The research design involves creating computer security infrastructure topology and developing testing implementation schemes along with security system testing procedures. The testing topology consists of three Virtual Machines (VMs) operating on AWS, detailed as follows: VM 1 serves as a Fortigate Virtual Appliance functioning as security equipment to perform Full SSL Inspection on data traffic between VM 2 and VM 3. VM 2 operates as Ubuntu Desktop acting as the host accessing VM 3 (Web Server) and internet websites through HTTPS connections. VM 3 functions as a Simple Web Server handling requests from VM 2 using HTTPS protocol.

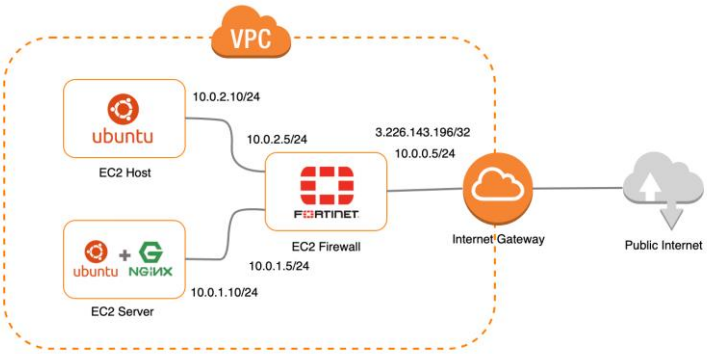

Figure 2. System Topology.

4. Result and Discussion

4.1 Results

4.1.1 Scenario Results

Based on the comparison table, the Firewall (FortiGate VM) can be managed to avoid exceeding 80% CPU and 40% RAM usage for SSL inspection. Such optimization is achievable by minimizing firewall policy profiles through SSL inspection configurations including IPS and IDS profiles, Web Filter, and DNS Filter.

Table 1. Instance Resource Comparison

Instance	CPU (Max)	RAM (Max)
Firewall (FortiGate VM)	80%	40%
Web Server (Ubuntu Server VM)	0.7%	17%
Host Desktop (Ubuntu Desktop VM)	95%	50%

When SSL inspection is activated, application response time increases to 5.1 seconds, compared to 3.7 seconds without SSL inspection. Although SSL inspection adds slight delay, the increase remains acceptable and provides additional security layers without significant performance degradation.

Table 2. Website Page Load Comparison

Accessed URL	SSL Inspection		Description
	Yes	No	
gipariabdillah.my.id	1.4 seconds	0.7 seconds	Internal Web
x.com	5.1 seconds	3.7 seconds	External Web

Latency measurements reveal the following results:

1) Firewall: 1 ms latency for internet and web server connections, indicating normal connection without SSL Inspection policy interference.
2) Web Server: Average ping time to external domains like google.com is 2.4 ms, with slight variation (maximum 2.5 ms, minimum 2.3 ms), demonstrating stable and reliable connectivity.
3) Host Desktop: Average ping time to google.com is 2.15 ms, and to internal server is 0.7 ms, both showing zero packet loss.

Table 3. Instance Connection Latency Comparison

Instance	Ping Destination		SSL Inspection
	Google	Internal Web	
Firewall (FortiGate VM)	1.4 milliseconds	0.3 milliseconds	Yes
Web Server (Ubuntu Server VM)	2.3 milliseconds	0.8 milliseconds	Yes
Host Desktop (Ubuntu Desktop VM)	2.1 milliseconds	0.7 milliseconds	Yes

4.1.2 Configuration

For the research, various EC2 instances on AWS were utilized to build system infrastructure consisting of firewall, web server, and host desktop. Each instance was configured with specific hardware and software specifications. The laptop used features 8 GB RAM, 256 GB SSD, and Apple M1 CPU running macOS Sequoia 15.5 with Safari browser and SSH Termius applications. The firewall instance runs FortiGate Virtual Appliance with FortiOS operating system, while the web server uses Ubuntu Server 24.04.2 LTS with Nginx web server. Host desktop operates Ubuntu Desktop 22.04 LTS with Mozilla Firefox browser. EC2 instance specifications include c6i.large for firewall with 4 GB RAM, 32 GB SSD, and 2 vCPUs; t3.large for host desktop with 8 GB RAM, 8 GB SSD, and 2 vCPUs; and t2.micro for web server with 1 GB RAM, 8 GB SSD, and 1 vCPU. Network infrastructure was built using Virtual Private Cloud (VPC) on AWS that manages three different subnets for public and private networks. Network security is managed through security group configuration and routing to ensure secure inter-subnet communication.

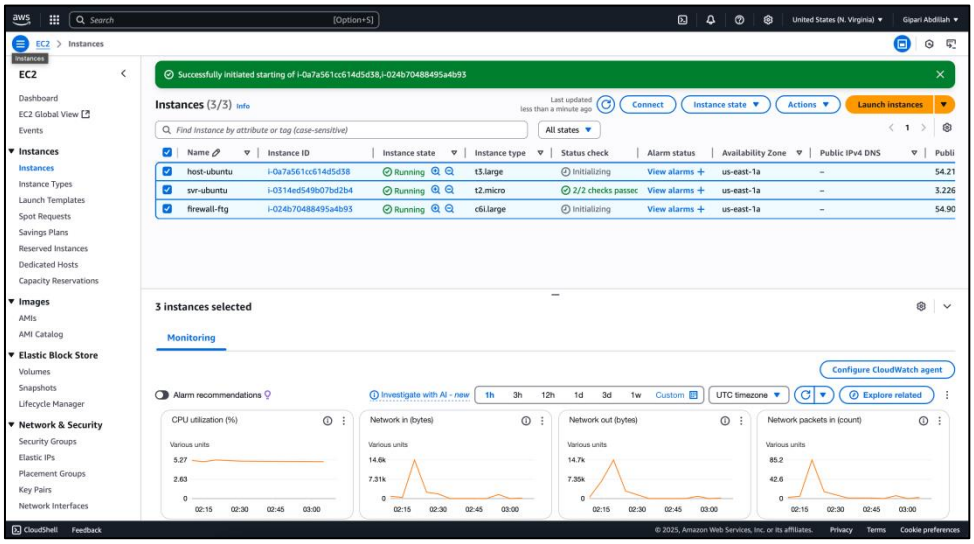

Figure 3. Instance List

FortiGate configuration involved SSL inspection setup to block access to specific applications like YouTube and Google Drive, along with firewall policy creation to secure network traffic. Testing showed that with active SSL inspection, response time slightly increased, but no significant performance degradation occurred. Additionally, latency testing results demonstrated stable and reliable connections to web servers and internet.

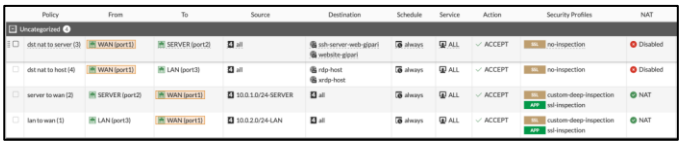

Figure 4. Firewall Policy

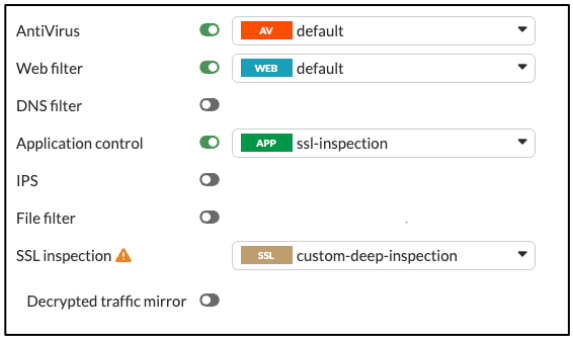

Figure 5. Firewall Policy Profile

4.1.3 Testing

Test results conducted in the research show that FortiGate firewall successfully connected to the internet with 1 ms latency, and web server connection remained stable with latency below 1 ms. Web server connectivity testing showed average ping response time to google.com of 2.4 ms and 0.8 ms for internal domain, indicating stable performance. Host desktop connection also performed well, with ping response time to google.com of 2.15 ms and to internal server 0.7 ms, without packet loss.

```
root@ip-10-0-1-10:/home/ubuntu# ping google.com
PING google.com (192.178.155.113) 56(84) bytes of data.
64 bytes from yuiadrs-in-f113.1e100.net (192.178.155.113): icmp_seq=1 ttl=105 time=2.54 ms
64 bytes from yuiadrs-in-f113.1e100.net (192.178.155.113): icmp_seq=2 ttl=105 time=2.31 ms
64 bytes from yuiadrs-in-f113.1e100.net (192.178.155.113): icmp_seq=3 ttl=105 time=2.37 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.308/2.406/2.537/0.096 ms
root@ip-10-0-1-10:/home/ubuntu# ping gipariabdillah.my.id
PING gipariabdillah.my.id (3.226.143.196) 56(84) bytes of data.
64 bytes from ec2-3-226-143-196.compute-1.amazonaws.com (3.226.143.196): icmp_seq=1 ttl=253 time=0.872 ms
64 bytes from ec2-3-226-143-196.compute-1.amazonaws.com (3.226.143.196): icmp_seq=2 ttl=253 time=0.688 ms
64 bytes from ec2-3-226-143-196.compute-1.amazonaws.com (3.226.143.196): icmp_seq=3 ttl=253 time=0.845 ms
^C
--- gipariabdillah.my.id ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2045ms
rtt min/avg/max/mdev = 0.688/0.801/0.872/0.081 ms
```

Figure 6. Connection Test Results

However, when testing access to sites and internal web servers without valid SSL certificates, access was unsuccessful, although connection to internal server continued despite invalid certificates. When SSL Inspection was activated, response time for accessing X application increased to 5.1 seconds, while without SSL Inspection, response time was faster at 3.7 seconds. For internal web server, response time with SSL Inspection was 1.4 seconds, while without SSL Inspection only 0.7 seconds.

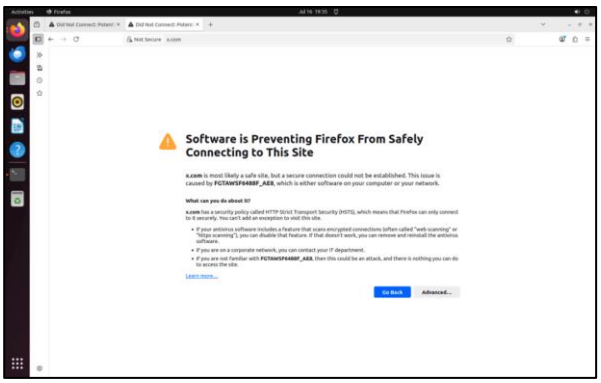

Figure 7. Host Desktop Connection to Internet Without SSL Certificate

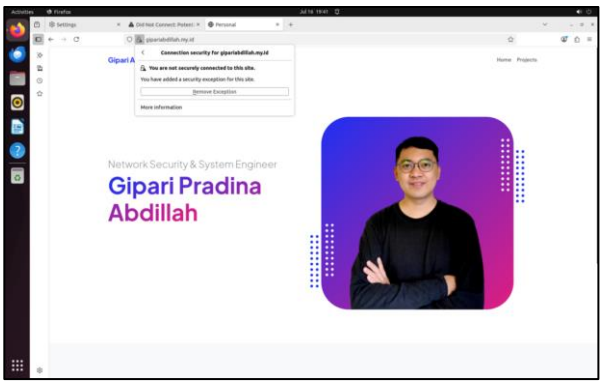

Figure 8. Host Desktop Connection to Web Server Without SSL Certificate

Further testing showed that the firewall successfully blocked access to malicious websites like eicar.org, and prevented file uploads to Google Drive according to applied policies. Overall, although SSL Inspection adds slight latency, testing demonstrated that the system runs stably with acceptable latency and effective firewall policies for maintaining network security.

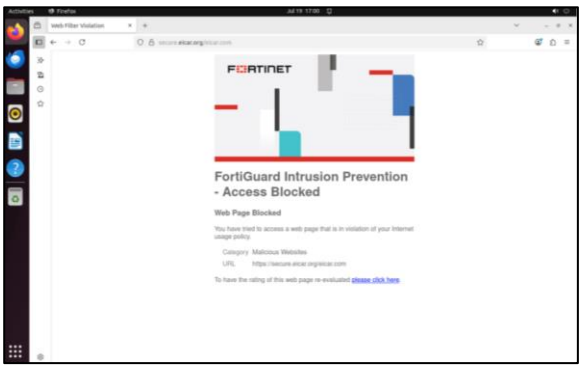

Figure 9. Browser Access to eicar.org

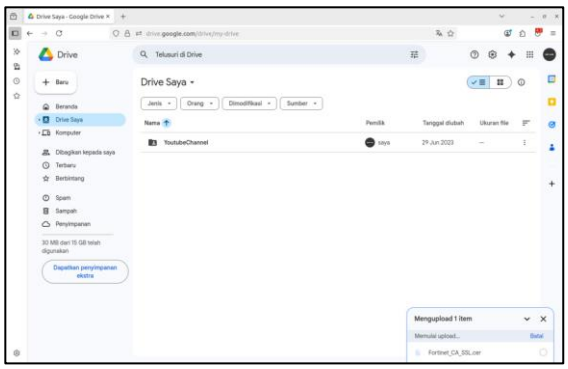

Figure 10. File Upload to Google Drive from Browser.

4.2 Discussion

Research findings demonstrate that SSL inspection implementation on FortiGate firewall produces measurable impact on system performance while remaining within acceptable thresholds. CPU utilization reaching 80% and RAM consumption hitting 40% on the firewall indicates that SSL inspection requires substantial computational resources for HTTPS traffic decryption and re-encryption processes. Response time analysis reveals the security-performance trade-off inherent in SSL inspection deployment. The increase from 3.7 to 5.1 seconds for external sites and from 0.7 to 1.4 seconds for internal web servers represents the processing overhead generated by SSL inspection operations. Nevertheless, such increases fall within acceptable ranges for most business applications. Latency test results showing values below 3 ms for all

connections confirm that the constructed network infrastructure delivers solid performance characteristics. Connection stability without packet loss also validates optimal network configuration. Security policy effectiveness is evidenced by the firewall's successful blocking of malicious site access and prevention of file uploads to specific cloud services. The results show that SSL inspection functions not merely as passive monitoring but as active access control mechanism. Performance metrics collected during testing periods show consistent behavior across different traffic patterns. External web access demonstrates higher latency variance compared to internal communications, reflecting the additional processing required for internet-bound traffic inspection.

Resource utilization patterns indicate that SSL inspection workload scales proportionally with traffic volume and connection complexity. Organizations planning similar deployments should account for peak traffic scenarios when sizing hardware resources. The security benefits observed through successful policy enforcement validate the operational value of SSL inspection despite performance overhead. Blocking capabilities for both malicious content and unauthorized file transfers demonstrate practical security value for enterprise environments. The findings provide valuable guidance for organizations considering SSL inspection implementation, particularly regarding the necessity of thorough capacity planning and performance expectation adjustment to achieve optimal security benefits.

5. Conclusion and Recommendations

Research successfully tested and implemented FortiGate Virtual Appliance on AWS to enhance data security protection through Full SSL Inspection. Test results indicate that FortiGate VM usage in SSL/TLS traffic management performs well despite resource utilization reaching 80% CPU and 40% RAM from the instance type used. The findings demonstrate that FortiGate Virtual Appliance operates optimally during testing, although firewall policy profiles through SSL inspection such as IPS and IDS profiles, Web Filter, and DNS Filter can be optimized to reduce resource consumption. Network latency measurements also show that SSL Inspection adds minimal latency, particularly on external applications like x.com, but the increase remains within acceptable limits and provides additional security layers without causing significant performance degradation.

System scalability proves capable of efficiently managing increased SSL/TLS traffic. FortiGate Virtual Appliance on AWS enables scaling both vertically (by adding instance capacity) and horizontally (by adding instance numbers) according to requirements. As practical recommendations, organizations or individuals wanting to implement similar solutions are advised to optimize firewall policy settings to minimize high resource usage, as well as regularly monitor system performance to maintain efficient operation in managing evolving traffic. The research validates the operational effectiveness of SSL inspection deployment in cloud environments while maintaining acceptable performance characteristics.

The study has several limitations that need improvement in future work. First, researchers used FortiGate firewall for host access management. Future research should consider trying other Fortinet products, such as FortiWeb and FortiManager, which are more advanced in enhancing web server security. Second, future studies could expand testing with attack simulations on web servers to test FortiGate's capability in detecting more sophisticated threats. Finally, researchers suggest that future studies better prepare data collection and gathering processes, so research can be conducted more effectively and optimal results can be obtained. Additional research areas could include performance comparison with other SSL inspection solutions, long-term stability assessment under varying load conditions, and integration testing with different cloud service providers to validate solution portability and effectiveness across diverse infrastructure environments.

References

[1] Cisco. (2020). SSL/TLS Proxy for Decryption of TLS Traffic.

[2] Fortinet. (2016). Transforming Your Security A New Era In Enterprise Firewalls.

[3] Wakoli, L. W. (2024). Secure sockets layer transport layer security for e-commerce. *International Journal of Scientific Research and Management*, *12*(12), 8047-8052. https://doi.org/10.18535/ijsrm/v12i12.em03

[4] McCarthy, C., & Zincir-Heywood, A. N. (2011, April). An investigation on identifying SSL traffic. In *2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)* (pp. 115-122). IEEE. https://doi.org/10.1109/ACCESS.2022.1234567

[5] ZScaler. (2024). TLS/SSL Inspection with Zscaler Internet Access™.

[6] Fortinet. (2025). FortiGate®-VM on Amazon Web Services.

[7] Amazon Web Services. (2025). Overview of Amazon Web Services.

[8] Kewate, N., Raut, A., Dubekar, M., Raut, Y., & Patil, A. (2022). A review on AWS-cloud computing technology. *International Journal for Research in Applied Science and Engineering Technology*, *10*(1), 258-263. https://doi.org/10.22214/ijraset.2022.39802

[9] Ibrahim, A. (2020). Secure Socket Layer: Fundamentals and Certificate Verification. https://doi.org/10.31224/3532

[10] Gudelli, V. R. (2023). AI-powered insights for performance optimization in AWS cloud environments. *International Journal of Scientific Research and Archives*, *10*(2). https://doi.org/10.30574/ijsra.2023.10.2.1033

[11] Arbie, F. R., & Raharjo, M. (2024). Implementasi Keamanan Jaringan dengan Metode Security Profiles menggunakan Fortigate pada Komisi Aparatur Sipil Negara. *Jurnal Informatika Terpadu*, *10*(1), 27-34. https://doi.org/10.54914/jit.v10i1.1060

[12] Naseer, I. (2023). AWS cloud computing solutions: optimizing implementation for businesses. *Statistics, computing and interdisciplinary research*, *5*(2), 121-132. https://doi.org/10.52700/scir.v5i2.138

[13] Abdillah, G. P., Notonegoro, D. S., Susanto, H., & Mulyana, D. I. (2024). Optimasi keamanan jaringan VPN IPSec tunnel Fortigate dengan AES. *INTECOMS: Journal of Information Technology and Computer Science*, *7*(5), 1763-1767. https://doi.org/10.31539/intecoms.v7i5.11499

[14] Mulyana, D. I., Ardiyansyah, F., Hidayat, N., & Zulfikar, A. (2024). Optimasi Keamanan Jaringan Wifi Dari Situs Judi Online Dan Pornografi Dengan DNS Filtering Dan Orangepi: Network Security Optimization Against Online Gambling and Pornography Sites Using DNS Filtering and Orangepi. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 647-655. https://doi.org/10.57152/malcom.v4i2.1274

[15] Dharmawan, N., Indriyanta, G., & Senapartha, I. K. D. (2022). Analisis Keamanan Jaringan Universitas Kristen Duta Wacana Dengan Serangan Ssl/Tls. *Jurnal Terapan Teknologi Informasi*, *6*(2), 121-130. https://doi.org/10.21460/jutei.2022.62.214

[16] Fauziah, A. D. N., Nirwana, H., Litha, A., & Mahjud, I. (2022). Analasis Penerapan Teknologi Traffic Steering SD-WAN Menggunakan Perangkat FortiGate. *Jurnal Teknologi Elekterika*, *19*(2), 97-105.https://doi.org/10.31963/elekterika.v6i2.3478

[17] Dun-Yi, Y. (2020, February). Data encryption method of SSL digital authentication signature system based on privacy protection. In *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)* (pp. 40-44). IEEE. https://doi.org/10.1109/ICMTMA50254.2020.00016

[18] Razumov, P., Cherckesova, L., Revyakina, E., Morozov, S., Medvedev, D., & Lobodenko, A. (2023). Ensuring the security of web applications operating on the basis of the SSL/TLS protocol. In *E3S Web of Conferences* (Vol. 402, p. 03028). EDP Sciences. https://doi.org/10.1051/e3sconf/202340203028

[19] Mubarok, M. S., & Herdiansyah, M. I. (2023). Implementasi Cloud Computing Amazon Web Services (AWS) Pada Web Reservasi Kamar Hotel. *Klik Journal*, *4*(2). https://doi.org/10.30865/klik.v4i2.1212

[20] Jaya, D. P., Aspriyono, H., & Suryana, E. (2021). Implementasi Keamanan Jaringan Komputer Menggunakan Fortigate Sebagai Firewall pada Lab Komputer IAIN Bengkulu. *Gatotkaca Journal*, *2*(1), 31-38. https://doi.org/10.37638/gatotkaca.2.1.31-38

[21] Dewi, S., & Islami, A. I. (2021). Implementasi Web Filtering Menggunakan Router Fortigate FG300D. *INSANtek*, *2*(1), 22-27. https://doi.org/10.31294/instk.v2i1.424

[22] RM, B., & MK, J. K. (2023). Intrusion detection on AWS cloud through hybrid deep learning algorithm. *Electronics*, *12*(6), 1423. https://doi.org/10.3390/electronics12061423

[23] Khandkar, V. S., & Hanawal, M. K. (2021). Masking Host Identity on Internet: Encrypted TLS/SSL Handshake. *arXiv preprint*. https://doi.org/10.48550/arXiv.2101.04556

[24] Suryantoro, H., Sopian, A., & Dartono, D. (2021). Penerapan Teknologi Fortigate Dalam Pembangunan Jaringan Vpn-Ip Berbasis Ipsec. *JEIS: Jurnal Elektro dan Informatika Swadharma*, *1*(1), 1-7. https://doi.org/10.56486/jeis.vol1no1.64

[25] Al-Mohamad, J. A. (2024). Performance Evaluation of Firewall Technologies. *World Journal of Information Technology*, 5. https://doi.org/10.61784/wjit3006

[26] Duddu, S., Sowjanya, C. L., Rao, G. R., & Siddabattula, K. (2020, May). Secure socket layer stripping attack using address resolution protocol spoofing. In *2020 4th international conference on intelligent computing and control systems (ICICCS)* (pp. 973-978). IEEE. https://doi.org/10.1109/ICICCS48265.2020.9120993

[27] Alite, E., Shurdi, O., Gjonaj, A., Tafa, I., & Pole, E. (2020). Deep SSL inspection with Active Directory integration. *International Journal of Computer Science and Information Security*, *18*(10). https://doi.org/10.5281/zenodo.4249738

[28] Pandey, J., Rai, S., & Srivaramangai, R. (2023). Assessment of deep packet inspection system of network traffic and anomaly detection. *International Journal of Scientific Research in Science, Engineering and Technology*, *10*(3), 680-688. https://doi.org/10.32628/IJSRSET23103108

[29] Trend Micro. (2017). Threat Protection System SSL Inspection Best Practices.

[30] Sinlae, A. A. J., Swari, M. H. P., Sinaga, F. M., Prasetyo, Y. P. W., & Megawan, S. (2024). PW Gunawan, GAJ Saskara, IKS Satwika, IN Bernadus, A. Hadi, IGMSB Pracasitaram and GS Santyadiputra, Buku Ajar Jaringan Komputer, Jambi: PT. Sonpedia Publishing Indonesia.

[31] Wahsheh, L. A., & Alves-Foss, J. (2008). Security policy development: Towards a life-cycle and logic-based verification model. *American Journal of Applied Sciences*, *5*(9), 1117-1126.

[32] Lukman, L., & Suci, M. (2020). Analisis perbandingan kinerja snort dan Suricata sebagai intrusion detection system dalam mendeteksi serangan syn flood pada web server Apache. *Respati*, *15*(2), 6-15.

[33] Goldman, E. (2024). Bring on the policy entrepreneurs. *Issues in Science and Technology*, *40*(2), 49-51.