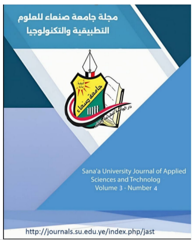
SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/


مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا
Sana'a University Journal of Applied Sciences and Technology
Volume 3 · Number 4
http://journals.su.edu.ye/index.php/jast

# Exploiting ASDM Vulnerabilities and Proactive Defense Strategies Against Sophisticated Malware in Cisco ASA Firewalls Security

**Malek Algabri**[1], **Jamil Rashid**[2] **and Ehab Al-junid** [2*]

[1]**Department of computer Science, Faculty of Computer and information technology, Sana'a University, Sana'a, Yemen,**
[2]**Department of Information Security, Faculty of Engineering and Information Technology, Emirates International University, Sana'a, Yemen.**

*Corresponding author: ehabw898@gmail.com

## Abstract

As cyber threats continue to evolve in sophistication, this study investigates critical vulnerabilities within Cisco ASA firewalls, with a particular emphasis on the Adaptive Security Device Manager (ASDM) interface. The research demonstrates how advanced malware can exploit the file-based architecture of ASDM to circumvent traditional firewall protections. A controlled virtual environment was established using VMware, incorporating a native Cisco ASA 5512-X firewall running software version 8.6(1), a Windows 10-based victim machine, and a Kali Linux attacker system.

To facilitate the exploitation process, substantial modifications were applied to the open-source tool **"theway"**, enabling it to extract and reassemble the .bin installation package without altering the structure of its internal components. Additionally, **7-Zip** was employed to unpack the embedded .msi installer files, allowing for the insertion of malicious payloads. The **WiX Toolset**, used in its unmodified state, was then utilized to rebuild the .msi packages. These .msi files were selected as the primary attack vector due to their broad compatibility across ASDM versions and reduced likelihood of detection. In contrast, attempts to inject unsigned Java .class files into the pdm.sgz archive were unsuccessful, as recent ASDM versions enforce strict digital signature verification.

The attack scenarios simulated in this study included reverse shell injection and credential interception through a man-in-the-middle (MITM) attack. Results revealed significant weaknesses in Cisco ASA's file validation procedures and outbound traffic monitoring mechanisms. To address these issues, a comprehensive multi-layered defense strategy is proposed, incorporating SIEM/EDR integration, robust digital signature enforcement, and internal network segmentation. The findings of this research contribute to both academic and practical efforts to enhance the security posture of enterprise firewall deployments against sophisticated malware threats.

## Article Info

**Keywords:**
Firewall Bypass, Malware Injection, MITM Attack, Reverse Shell Attack, WiX Toolset, Adaptive Security Device Manager (ASDM), MSI File Manipulation.

**Article History:**
**Received:** 26-May-2025,
**Revised:** 20-June-2025,
**Accepted:** 15-July-2025,
**Available online:** 28 August 2025.

## 1. INTRODUCTION

As digital infrastructure continues to permeate all aspects of modern life, ensuring the confidentiality, integrity, and availability of network systems has become a fundamental priority for organizations. One of the cornerstone technologies in defending against cyber threats is the firewall, which serves as a critical gatekeeper between internal trusted networks and untrusted external sources, such as the Internet. The Cisco Adaptive Security Appliance (ASA) is one of and government sectors [1]. Its comprehensive feature set, including packet filtering, VPN support, and integrated intrusion detection/prevention systems (IDS/IPS)—combined with high configurability, makes it a preferred choice in complex network environments [2]. However, despite its advanced security mechanisms, the Cisco ASA is not impervious to exploitation. Multiple vulnerabilities have surfaced in recent years, some of which have exposed critical flaws in internal services and configuration logic. A particularly sensitive component is the Adaptive Security Device Manager (ASDM), which is a graphical interface used to manage ASA configurations via a local web interface. The ASDM's role as a trusted administrative tool makes it an attractive target for adversaries to infiltrate network defenses. This study investigates how malicious actors can exploit ASDM to execute sophisticated attacks, such as reverse shell injections and man-in-the-middle (MITM) exploits [3]. While these attack techniques are not novel in themselves, their specific application to Cisco ASA, particularly through ASDM file manipulation, remains largely underexplored in academic and practical domains. This gap forms the foundation of this study's originality and relevance.

In addition to emerging attack vectors, Cisco ASA devices have historically been vulnerable to a range of threats, including dDenial-of-sService (DoS) attacks, remote code execution (RCE), unauthorized configuration changes, and remote access violations. These categories highlight the risks of relying solely on default configurations and underscore the need for comprehensive security assessments and proactive defense strategies.

## 2. RESEARCH CONTRIBUTION

This study offers several key contributions that enhance both theoretical understanding and practical defense mechanisms in the context of Cisco ASA firewalls:

### 2.1. Discovery of Novel Vulnerabilities in ASA File Validation Mechanisms

The reveal critical flaws in Cisco ASA's internal file verification system, particularly in its insufficient enforcement of digital signatures [4]. Injection of malware into the system. msi and .bin files using customized tools such as the WiX Toolset and a modified version of "theway" [5], the study demonstrated that malicious files could bypass integrity checks, revealing a fundamental design weakness.

### 2.2. Practical Simulation of Sophisticated Attack Vectors

Through real-world simulations of reverse shell and man-in-the-middle (MITM) attacks in a controlled virtual lab, the study revealed the ASA firewall's inability to detect or prevent threats that target internal system components, such as ASDM, even when the system is fully updated [6].

### 2.3. Development of an Unconventional Attack Model

Unlike conventional methods that rely solely on exploiting software bugs, this study introduces an innovative model that manipulates the structural design of legitimate Cisco installation files. This approach broadens the scope of attack analysis by examining the file-based infiltration paths that are often overlooked.

### 2.4. Proposal of a Comprehensive Security Framework

The research presents a set of actionable security enhancements, including

· Integration with endpoint detection and response (EDR) tools is based on behavioral analysis [7].

· Reinforcement of outbound traffic policies and digital signature enforcement.

· Adoption of a Defense-in-Depth model featuring internal segmentation and real-time intrusion detection.

### 2.5. Contribution to Academic Literature and Industry Practice

The findings are documented in a detailed white paper that not only serves as a scholarly reference, but also offers scalable insights applicable to other firewall systems with similar architectural designs.

Collectively, these contributions bridge the gap between theory and application, equipping researchers and practitioners with deeper insights and practical methodologies for enhancing firewall resilience.

## 3. PROBLEM STATEMENT

Cisco ASA firewalls are widely deployed in enterprise networks as primary defense mechanisms against external threats. The Adaptive Security Device Manager (ASDM), a core component of their management interface, relies on the installation packages in .bin and .msi format for deployment and update. However, a critical security concern arises from insufficient validation and verification of these installation files.

The ASA platform fails to enforce strong integrity checks and digital signature validations on these files, making it possible for adversaries to inject malicious code into legitimate ASDM installation packages without detection. Additionally, the ASA lacks a comprehensive outbound traffic inspection, which allows reverse shell connections to be established from compromised systems without raising alarms. This combination of weaknesses presents a high-risk scenario in which trusted administrative tools become vector for malware deployment [8].

This study seeks to address this gap by simulating realistic attack scenarios that exploit these vulnerabilities and by analyzing ASA's default behavior of the ASA in response. The objective was to demonstrate how file-based attacks can bypass existing security controls and propose effective mitigation strategies.

## 4. TECHNICAL CHALLENGES

Throughout the execution of this research, several technical challenges emerged that impacted the design and implementation of the simulated attack scenarios. One of the primary difficulties was the inability to execute unsigned Java. class files in the ASDM environment. Modern versions of the Cisco ASDM strictly enforce digital signature verification, preventing the successful injection of custom Java payloads into the pdm.sgz archive. This limits the ability to explore client-side exploits through traditional Java-based methods.

Another major challenge involved the handling of Cisco's proprietary file structures, particularly ASDM . bin installation packages. Publicly available tools, such as theway and Getchoo, were not fully compatible with the internal packaging format used by Cisco. This required significant modifications to the open-source method to enable proper extraction and repackaging of bin files while preserving the file integrity.

In addition, payloads are injected into . msi installers pose compatibility and validation issues. Manual unpacking with 7-Zip, combined with rebuilding the packages using the WiX Toolset, requires careful structuring to ensure that tampered installers would still be accepted and executed by the ASDM installer without errors or crashes.

These challenges highlight the technical complexity of exploiting ASDM in a realistic manner, and underscore the necessity of deep protocol understanding, custom tool development, and iterative testing in cybersecurity research involving real-world enterprise-grade systems.

## 5. RESEARCH OBJECTIVES

This study was designed with the following core objectives:

**1.** To analyze the internal architecture of Cisco ASA firewalls, focusing on their operational mechanisms and packet-handling logic [9].

**2.** To assess the effectiveness of ASA firewalls in detecting and mitigating advanced malware threats by examining both real-world and simulated attack scenarios in whichere traditional defenses awere bypassed [10].

**3.** To simulate reverse shell-based attack scenarios in order to demonstrate how adversaries can craft and deploy malware capable of evading the ASA's built-in security validations [11].

**4.** To conduct man-in-the-middle (MITM) simulations that reveal how authentication credentials can be intercepted within a network protected by Cisco ASA.

**5.** To propose practical solutions and technical countermeasures aimed at strengthening the ASA's overall security posture, including configuration enhancements and integration with tools such as EDR and SIEM systems [12].

**6.** To raise awareness about the limitations of relying solely on firewalls as a security perimeter, and to emphasize the need for a layered defense approach incorporating internal segmentation, behavioral analysis, and real-time monitoring.

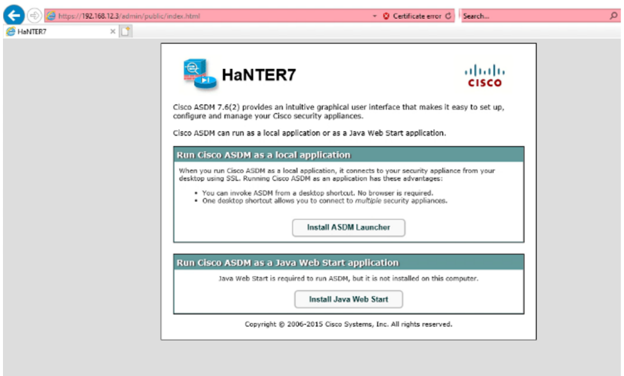
## 6. EXPERIMENTAL SCENARIO AND ATTACK SIMULATIONS

The test environment featured a Cisco ASA 5512-X firewall running software version 8.6(1)2, which reflects a commonly deployed configuration in real-world enterprise environments. This version was chosen because of its compatibility with the ASDM interface and exposure to exploitable vulnerabilities through installation file manipulation.

To evaluate the real-world applicability of the proposed attack models, a virtual testbed was designed and implemented using VMware [13]. The environment consisted of two virtual machines: one emulating a typical victim system running Windows 10, and the other serving as the attacker platform running Kali Linux.

Both machines were connected to a real Cisco ASA firewall through a managed network environment, configured with default settings to mirror the typical deployment scenarios.

Within this setup, a series of attack simulations was conducted to test the response of the firewall to advanced malware techniques.



Cisco ASDM 7.6(2) provides an intuitive graphical user interface that makes it easy to set up, configure and manage your Cisco security appliances.

Cisco ASDM can run as a local application or as a Java Web Start application.

Run Cisco ASDM as a local application

When you run Cisco ASDM as a local application, it connects to your security appliance from your desktop using SSL. Running Cisco ASDM as an application has these advantages:

- You can invoke ASDM from a desktop shortcut. No browser is required.
- One desktop shortcut allows you to connect to *multiple* security appliances.

Run Cisco ASDM as a Java Web Start application

Java Web Start is required to run ASDM, but it is not installed on this computer.

Copyright © 2006-2015 Cisco Systems, Inc. All rights reserved.

**Figure 1.** Modified HTML interface within the ASDM installation package displaying a custom message, demonstrating successful injection and tampering with the installer's visual components.

These simulations were divided into two main scenarios: reverse shell exploitation through modified installation files, and a man-in-the-middle (MITM) attack targeting credential interception.

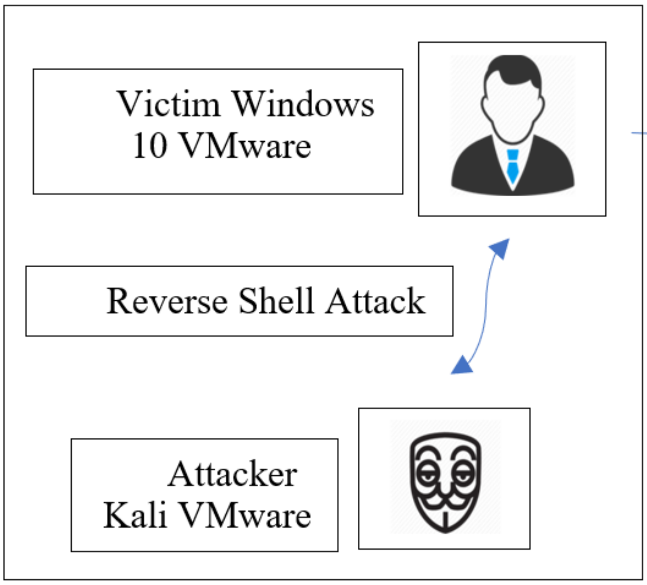
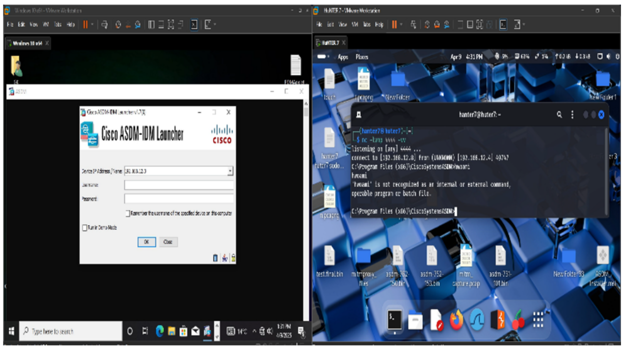


**Figure 2.** Visualization of the reverse shell attack where the victim's execution of a tampered ASDM installer results in a command-line session with the attacker.

## 6.1. Reverse Shell Exploitation Scenario

The reverse shell attack simulation commenced with the use of a custom-modified tool named *theway*, which was employed to extract, alter, and recompile. bin installation file used by the Cisco ASA. The process involved analyzing the internal manifest and injecting a malicious Java .class file into the pdm.sgz archive, which serves as the graphical interface for the ASDM.



```
┌──(hanter7@huter7)-[~]
└─$ nc -lvnp 4444 -vv
listening on [any] 4444 ...
connect to [192.168.12.8] from (UNKNOWN) [192.168.12.4] 49747
C:\Program Files (x86)\CiscoSystems\ASDM>hwoami
hwoami
'hwoami' is not recognized as an internal or external command,
operable program or batch file.

C:\Program Files (x86)\CiscoSystems\ASDM>
```

**Figure 3.** Successful reverse shell connection established between the victim machine and the attacker using a modified ASDM installer file.

However, this approach was rendered ineffective because of Java's strict signature enforcement, which prevented the execution of unsigned classes within the ASDM environment. During the initial stage of the experiment, an attempt was made to inject unsigned Java. class file into the pdm.sgz archive, which forms a part of the graphical interface of the ASDM. The objective was to exploit ASDM's internal Java execution environment to establish a reverse shell session.

However, the approach failed because of **Cisco ASA's strict enforcement of digital signature validation for Java files**. The ASA platform uses a **Java security model** that restricts the execution of any unsigned or self-signed Java classes embedded in the ASDM components. As a result, the injection. class file was rejected at runtime, and the graphical interface failed to be loaded properly.

In addition, this method is only applicable to **older versions of ASDM**, where digital signature verification is either absent or loosely implemented. Modern versions explicitly enforce code signing using certificates issued by Cisco or trusted Certificate Authorities (CAs), effectively blocking any unauthorized modification of Java archives.

This limitation highlights the need to shift toward more viable attack vectors, leading to the adoption of . msi-based payload injection, which bypasses signature checks at a different levels of the ASDM deployment pipeline.

In response to this limitation, the methodology was refined to focus on a more viable vector that embeds legitimate malware. msi installation files, specifically asdm50-install. msi and dm-launcher.msi. Tools such as 7-Zip and the WiX Toolset have been utilized to extract, modify, and rebuild MSI packages while preserving their original structure [14].

The modified MSI files were reintegrated into . bin package using a refined version of *theway*. Upon deployment to the ASA firewall and subsequent download via a victim machine, the malicious installer executed successfully, establishing an outbound reverse-shell connection to the attacker's host.

Notably, no alerts or warnings were generated by the

**Table 1.** Summary of Simulated Attack Scenarios and Their Outcomes

| Attack Scenario | Tools Used | Target Component | Outcome |
|---|---|---|---|
| Reverse Shell via MSI Injection | WiX Toolset, 7-Zip, theway | asdm50-install.msi | √ Reverse connection established; ASA failed to detect the threat |
| Java Class Injection (ASDM GUI) | Getchoo, modified theway | pdm.sgz | ☒ Execution blocked; ASA rejected unsigned Java class |
| MITM via SSL/TLS Exploitation | MITMweb | ASDM login portal | √ Credentials intercepted; ASA failed to enforce certificate validation |

ASA during this process, indicating a critical flaw in its file integrity verification and outbound connection monitoring capabilities.

The outcomes illustrated in Table 1 reflect the varying levels of susceptibility within the Cisco ASA architecture when subjected to targeted attack scenarios. Notably, reverse shell injection via MSI files was the most effective method, fully bypassing ASA signature validation and traffic filtering. While the Java class injection was unsuccessful owing to signature enforcement, the MITM attack proved the inadequacy of the ASA's SSL/TLS configurations in protecting credential exchanges. These findings reinforce the need for a defense-in-depth approach that addresses vulnerabilities across system configurations and communication protocols.

#### *6.1.1. Toolchain Customization and Workflow Overview*

To successfully craft and deploy the malicious ASDM installer, a chain of tools was used each serving a distinct role in the attack simulation process:

· **theway (Modified Version)**

Originally developed for unpacking and repackaging . bin files, this tool requires substantial modifications to suit the ASDM structure of Cisco. The original version created entirely new package structures rather than preserving and rebuilding the existing ones. The customized version was enhanced for extraction . bin contents as is, and reassemble them accurately without altering the metadata or directory layout. It was used both at the beginning to extract the ASDM installation bundle and at the end—to rebuild the final . bins after modification.

· **7-Zip:**

After extraction .bin file using this method, **7-Zip** was used to manually unpack the embedded file. msi installer files (e.g., asdm50-install. msi). This allowed full access to the internal file structure of the MSI package, where malicious payloads (such as reverse- shell scripts) were added to specific directories within the installer.

· **WiX Toolset (Standard Use)**

Once malicious components were added, the WiX Toolset was used to recompile and regenerate . msi file, preserving its original structure and appearance. Notably, no changes were made to the tool. The . The msi vector was selected over Java-based injection because of its compatibility with all ASDM versions and lower detection risk. Unlike .class injection into pdm.sgz, which is restricted to older ASDM builds and requires unsigned Java execution (which is blocked in most modern versions). Thus, the msi method proved to be more reliable.

This multi-tool pipeline enabled the successful creation of a tampered ASDM installer that retained Cisco's expected file formats and structures while embedding functional malware. The integration of modified and standard tools was key to bypassing Cisco ASA file validation routines, highlighting a critical weakness in its internal package handling mechanisms.

**Unlike .bin file tampering, the .msi-based method offered broader compatibility and easier bypassing of digital signature checks, making it the more reliable vector.**

## 6.2. MITM Exploitation Scenario

The second simulation focused on exploiting man-in-the-middle (MITM) vulnerabilities to compromise the sensitive authentication data transmitted within a Cisco ASA–protected network. The objective was to assess whether improperly configured encryption protocols [15] could be leveraged to intercept user credentials without triggering security mechanisms.

In this scenario, the attacker, positioned within the internal network, utilize the MITMweb tool to exploit weaknesses in outdated or misconfigured SSL/TLS settings.

These vulnerabilities allow the attacker to intercept HTTP-based authentication traffic between internal users and the ASDM management interface of the ASA.

The attack was executed by spoofing the firewall's certificate and establishing a transparent proxy to relay and decrypt encrypted credentials in real time. Consequently, sensitive information such as usernames and passwords was clearly captured without any detection from the ASA device.

This experiment underscores the critical importance of enforcing modern encryption protocols (e.g., TLS 1.2 or higher) [16], disabling insecure legacy protocols (SSL, TLS 1.0/1.1)., and employing certificate validation and multi-factor authentication to mitigate MITM risks.

It should be emphasized that the attack simulations

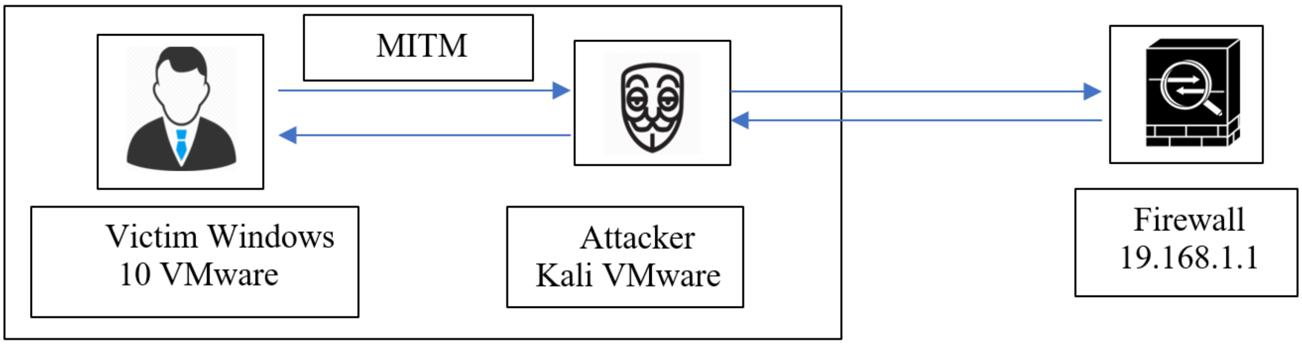
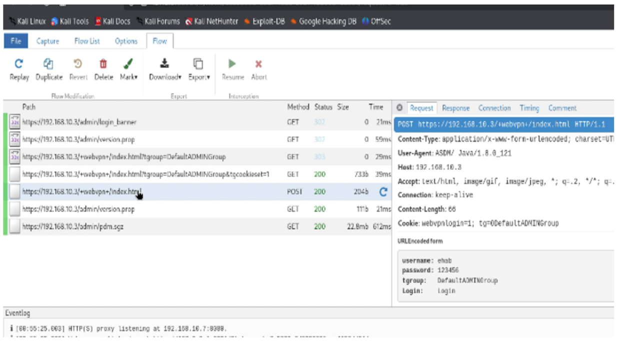


**Figure 4.** Graphical representation of the MITM attack where the attacker intercepts ASDM login credentials over a spoofed TLS connection.



```
Path                                                                     Method Status Size   Time
https://192.168.10.3/admin/login_banner                                  GET    302    0      21ms
https://192.168.10.3/admin/version.prop                                  GET    302    0      59ms
https://192.168.10.3/+webvpn+/index.html?tgroup=DefaultADMINGroup        GET    303    0      29ms
https://192.168.10.3/+webvpn+/index.html?tgroup=DefaultADMINGroup&tgcookieset=1 GET 200 733b 39ms
https://192.168.10.3/+webvpn+/index.html                                 POST   200    204b
https://192.168.10.3/admin/version.prop                                  GET    200    111b   21ms
https://192.168.10.3/admin/pdm.sgz                                       GET    200    22.8mb 612ms
```

```
POST https://192.168.10.3/+webvpn+/index.html HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=UT
User-Agent: ASDM/ Java/1.8.0_121
Host: 192.168.10.3
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.
Connection: keep-alive
Content-Length: 66
Cookie: webvpnlogin=1; tg=0DefaultADMINGroup

username: ehab
password: 123456
tgroup:   DefaultADMINGroup
Login:    login
```

```
[08:55:25.663] HTTP(S) proxy listening at 192.168.10.7:8080.
```

**Figure 5.** Captured ASDM login credentials during a man-in-the-middle (MITM) attack using MITMweb, illustrating successful interception of authentication data over an insecure or misconfigured SSL/TLS channel.

conducted in this study were based on a single controlled scenario designed to assess the feasibility and uncover exploitable weaknesses. The goal was not to produce statistically generalized results but rather to validate a focused proof-of-concept model that reflects real-world risk exposure.

It is important to note that the simulation experiments conducted in this study were based on a single, controlled environment. The goal was not to produce statistically generalized results, but rather to demonstrate a proof-of-concept model that reflects potential real-world risk exposure under specific conditions.

## 7. PROPOSED SOLUTIONS AND SECURITY ENHANCEMENTS

### 7.1. Countermeasures for Reverse Shell Exploitation

To mitigate the risks posed by reverse shell attacks initiated through manipulated ASDM installation files, the following technical countermeasures are recommended.:

**1. Integration with SIEM Platforms**

Cisco ASA should be integrated with a Security Information and Event Management (SIEM) systems such as Wazuh or Splunk. This allows for real-time monitoring of file uploads, configuration changes, and anomalous events within the ASA. Custom alerts should be configured to flag any attempt at uploading . bin or .msi files, particularly those that did not originate from trusted sources.

**2. Restricting Access to the ASDM Interface**

Limits administrative access by enforcing IP-based access control lists (ACLs). Only trusted subnets or dedicated management networks must be granted access. In addition, ASDM access via HTTP/HTTPS from public or external networks should be disabled.

**3. Egress Traffic Monitoring and Control**

Because reverse shell connections rely on outbound traffic initiated from a compromised host, implementing robust monitoring of egress connections is essential. Solutions such as NetFlow, Cisco Stealthwatch, or behavior-based detection tools should be used to identify abnormal traffic patterns. Access Control Lists (ACLs) should also be used to restrict communications with unknown or high-risk external IP addresses.

**4. File Integrity Verification and Digital Signature Enforcement**

All uploaded the ASDM and . bin files must be verified prior to installation. This includes enforcing SHA256 checksum validation and rejecting files that are not digitally signed by Cisco. Manual verification steps should be incorporated into the update procedures.

**5. Endpoint Detection and Response (EDR) for Internal Devices**

Because the reverse connection is initiated from a host device inside the network, deploying EDR solutions, such as CrowdStrike, Velociraptor, or similar tools, is essential. These tools can detect unusual behaviors, such as PowerShell or CMD-based remote connections, and can be configured to automatically send alerts or trigger isolation policies.

### 7.2. Countermeasures for Man-in-the-Middle (MITM) Exploitation

To prevent credential interception and unauthorized access resulting from MITM attacks targeting Cisco ASA environments [17], the following security measures should be implemented:

**1. Enforce Modern Encryption Protocols**

Only strong encryption standards such as TLS 1.2 or TLS 1.3 should be permitted. All legacy protocols, including SSL, TLS 1.0, and TLS 1.1, must be explicitly disabled on both the ASA and client systems to reduce susceptibility to downgrade attacks [18].

**2. Implement Strong Certificate Validation Practices**

Digital certificates used by the ASDM interface must be signed by a trusted Certificate Authority (CA). Self-signed or expired certificates were disallowed. Additionally, it enables strict certificate validation of all client systems accessing the firewall management interface.

**3. Enable Multi-Factor Authentication (MFA)**

Access to the ASDM interface should be protected using multi-factor authentication mechanisms. This adds an additional layer of security beyond password-based login, making it significantly more difficult for attackers to gain administrative access, even if the credentials are compromised.

**4. Adopt a Defense-in-Depth Security Architecture**

Organization should not rely solely on the ASA as a single point of defence. Deploying internal Intrusion Detection and Prevention Systems (IDPS), such as Snort or Suricata, can help monitor lateral movement and suspicious activities within the network [19]. In addition, network segmentation should be enforced using internal ACLs to restrict unnecessary communication flows.

**5. Establish a Dedicated Management Network (Out-of-Band)**

Administrative access to the ASA device should be isolated from the primary data network. Using a separate, secured out-of-band management network helps prevent attackers from reaching the ASDM interface, even if internal systems are compromised.

## 8. CONCLUSION

This study uncovered critical security weaknesses in the Cisco ASA firewalls, particularly within the ASDM management interface. Through controlled simulations of reverse shell and man-in-the-middle (MITM) attacks, it was demonstrated that adversaries could exploit deficiencies in file validation, digital signature enforcement, and outbound traffic monitoring. The ability to manipulate . msi files without detection, along with the interception of authentication credentials, highlights serious gaps in the platform's default security posture.

Additionally, the research emphasized the effectiveness of tool chain customization in circumventing built-in safeguards, ultimately enabling the deployment of successful, undetected payloads. In response, the study proposed a comprehensive multi-layered security framework that incorporates SIEM/EDR [20] integration, robust encryption enforcement, internal segmentation, and behavioral endpoint monitoring.

The findings offer both theoretical insights and practical value to the cybersecurity field, particularly in the context of securing enterprise firewall environments against sophisticated malware threats [21].

**Ultimately, this research successfully demonstrated how ASDM vulnerabilities can be leveraged in real-world attack scenarios and provided effective countermeasures that enhance the resilience of Cisco ASA deployments.**

## REFERENCES

[1] A. Parikh. “Cloud security and platform thinking: An analysis of Cisco Umbrella, a cloud-delivered enterprise security”. PhD thesis. Massachusetts Institute of Technology, 2019. URL: https://dspace.mit.edu/handle/1721.1/123456.

[2] A. Trisolino. “Analysis of Security Configuration for IDS/IPS”. PhD thesis. Politecnico di Torino, 2023. URL: https://webthesis.biblio.polito.it/id/eprint/25807.

[3] M. Conti, N. Dragoni, and V. Lesyk. “A survey of man-in-the-middle attacks”. In: *IEEE Commun. Surv. & Tutorials* 18.3 (2016), pp. 2027–2051. DOI: 10.1109/COMST.2016.2548426.

[4] S. Lekkala and P. Gurijala. *Security and Privacy for Modern Networks*. Springer Nature, 2024. URL: https://link.springer.com/book/10.1007/978-981-99-1567-3.

[5] B. I. Tuleuov and A. B. Ospanova. *Beginning C++ Compilers*. Springer Nature, 2024. DOI: 10.1007/978-3-031-51863-4.

[6] J. E. Jaakkola and K. B. Drake. “ASDM: The universal systems development methodology”. In: *J. Syst. Manag.* 42.2 (1991), p. 6. URL: https://www.proquest.com/openview/174daa3c31e3b77fca99898b9c571d22.

[7] D. S. Deshpande et al. “Endpoint detection and response system: Emerging cyber security technology”. In: *The International Conference on Intelligent Systems & Networks*. Springer, 2024, pp. 202–213. DOI: 10.1007/978-981-99-3651-7_16.

[8] Cybersecurity News. *Hackers exploit Microsoft Management Console to drop backdoor on Windows*. Accessed 2024. Jan. 2024. URL: https://cybersecuritynews.com/hackers-exploit-windows-management-console/.

[9] M. Uzsoki. “Remote management of a software-defined network with mobile application using REST API of Cisco APIC-EM controller”. PhD thesis. University of Applied Sciences Technikum Wien, 2020. URL: https://phaidra.boku.ac.at/o:11683.

[10] Ö. Aslan et al. “A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions”. In: *Electronics* 12.6 (2023), p. 1333. DOI: 10.3390/electronics12061333.

[11] S. Vemula, J. Gooley, and R. Hasan. *Cisco Software-Defined Access*. Cisco Press, 2020. URL: https://www.pearson.com/en-us/subject-catalog/p/program/P2programid/9780136448389.

[12] R. Boddu and S. Lamppu. *Microsoft Unified XDR and SIEM Solution Handbook: Modernize and Build a Unified SOC Platform for Future-Proof Security*. Packt Publishing, 2024. URL: https://www.packtpub.com/product/microsoft-unified-xdr-and-siem-solution-handbook/9781804613893.

[13] J. Sendorek, T. Szydlo, and R. Brzoza-Woch. “Software-defined virtual testbed for IoT systems”. In: *Wirel. Commun. Mob. Comput.* (2018), Article ID 1068261. DOI: 10.1155/2018/1068261.

[14] B. I. Tuleuov and A. B. Ospanova. *Beginning C++ Compilers*. Springer Nature, 2024. DOI: 10.1007/978-3-031-51863-4.

[15] National Institute of Standards and Technology. *Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations*. Tech. rep. SP 800-52 Rev. 2. U.S. Department of Commerce, 2020. URL: https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final.

[16] Firdaus AlHrazi, Malek Algabri, et al. "Hierarchical Blockchain as Line Defense of Attacks to Messages Propagation in VANET". In: *J. Appl. Sci. Technol.* 1.3 (2023). DOI: 10.59628/jast.v1i3.370.

[17] Firdaus AlHrazi, Malek Algabri, et al. "Hierarchical Blockchain as Line Defense of Attacks to Messages Propagation in VANET". In: *J. Appl. Sci. Technol. (JAST)* 1.3 (2023). DOI: 10.59628/jast.v1i3.370.

[18] National Institute of Standards and Technology. *Guidelines for the selection, configuration, and use of Transport Layer Security (TLS) implementations*. Tech. rep. SP 800-52 Rev. 2. U.S. Department of Commerce, 2020. URL: https://csrc.nist.gov/publications/detail/sp/800-52/rev-2/final.

[19] Malek Algabri, Abdulrahman A. Alsabri, Firdaus AlHrazi, et al. "Deep Learning and Blockchain for Detection and Prevention Abuse of Privileges". In: *2024 International Conference on Emerging Trends in Informatics (ICETI)*. 2024. DOI: 10.1109/ICETI63946.2024.10777144.

[20] D. S. Deshpande et al. "Endpoint detection and response system: Emerging cyber security technology". In: *The International Conference on Intelligent Systems & Networks*. Springer, 2024, pp. 202–213. DOI: 10.1007/978-981-99-3651-7_16.

[21] Ö. Aslan et al. "A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions". In: *Electronics* 12.6 (2023), p. 1333. DOI: 10.3390/electronics12061333.