WEB APPLICATION

GRAY BOX

PENETRATION TESTING REPORT

for

safunet.com

Table Of Contents

Executive Summary

This report presents the results of the “Gray Box'' penetration testing for the **safunet.com** WEB application. The recommendations provided in this report are structured to facilitate the remediation of the identified security risks. Evaluation ratings compare information gathered during the engagement to “best in class” criteria for security standards. We believe that the statements made in this document provide an accurate assessment of **safunet.com** current security as it relates to **safunet.com**'s data.
We highly recommend reviewing the Summary section of business risks and High-Level Recommendations for a better understanding of risks and discovered security issues.

Scope of assessment	https://safunet.com
Security Level	A
Grade	Excellent

Grading Criteria:

Grade	Security	Criteria Description
A	Excellent	The security exceeds “Industry Best Practice” standards. The overall posture was found to be excellent with only a few low-risk findings identified.
B	Good	The security meets accepted standards for “Industry Best Practice.” The overall posture was found to be strong with only a handful of medium- and low-risk shortcomings identified.
C	Fair	Current solutions protect some areas of the enterprise from security issues. Moderate changes are required to elevate the discussed areas to “Industry Best Practice” standards
D	Poor	Significant security deficiencies exist. Immediate attention should be given to the discussed issues to address exposures identified. Major changes are required to elevate to “Industry Best Practice” standards.
F	Inadequate	Serious security deficiencies exist. Shortcomings were identified throughout most or even all of the security controls examined. Improving security will require a major allocation of resources.

1.1 Project Objectives

Our primary goal within this project was to provide **safunet.com** with an understanding of the current level of security in the web application and its infrastructure components. We completed the following objectives to accomplish this goal:

- Identifying application-based threats to and vulnerabilities in the application
- Comparing **safunet.com's** current security measures with industry best practices
- Providing recommendations that **safunet.com** can implement to mitigate threats and vulnerabilities and meet industry best practices

The Common Vulnerability Scoring System (CVSS) version 3.0 was used to calculate the scores of the vulnerabilities found. When calculating the score, the following CIA provision, supplied by the **safunet.com** has been taken into account:

Scope	Confidentiality	Integrity	Availability
All scope objects	High	High	High

1.2 Scope & Timeframe

Testing and verification were performed between March 28, 2023 and April 17, 2023. This project's scope was limited to the **safunet.com** application and the specific infrastructure on which the application resides.

We conducted the tests using a non-production version of **safunet.com**. All other applications and servers were out of scope. All testing and verification were conducted from outside of **safunet.com** offices.

The following hosts were considered to be in scope for testing,

1.2.1 Links in scope

Scope:
https://safunet.com/

1.3 Summary of Findings

Our assessment of the **safunet.com** application revealed the following vulnerabilities.

Security experts performed manual security testing according to the OWASP Web Application Testing Methodology, which demonstrates the following results.

Severity	Critical	High	Medium	Low	Informational
Number of issues	0	0	0	0	0

Severity scoring:

- Critical – Immediate threat to key business processes.
- High – Direct threat to key business processes.
- Medium – Indirect threat to key business processes or partial threat to business processes.
- Low – No direct threat exists. The vulnerability may be exploited using other vulnerabilities.
- Informational – This finding does not indicate vulnerability, but states a comment that notifies about design flaws and improper implementation that might cause a problem in the long run.

The exploitation of found vulnerabilities may cause full compromise of some services, stealing users' accounts, and gaining organizations' and users' sensitive information.

1.4 Summary of Business Risks

In the case of **safunet.com** application

Critical severity issues can lead to

- Threat Actor Persistence on Your internal network
- Pivoting the network and obtaining critical information from internal-hosted databases
- Injection ransomware into the network system.

High severity issues can lead to:

- Complete decommissioning of the system.
- Customer's sensitive data disclosure.
- Takeovers of users' accounts and horizontal privilege escalation

Medium severity issues can lead to:

- Gaining the access to users session after the exploitation of high-level risks.
- Exploitation of users' computers and laptops

1.5 High-Level Recommendations

Taking into consideration all issues that have been discovered, we highly recommend to:

- Continuously monitor logs for anomalies to detect abnormal behavior and fraud transactions. Dedicate security operations engineer to this task
- Deploy Web Application Firewall solution to detect any malicious manipulations.
- Continuously inventory the versions of both client-side and server-side components (e.g. frameworks, libraries) and their dependencies.
- Review security configuration of all additional modules.
- Check publicly exposed endpoints on a web server and make sure that disclosed information is not sensitive. Debug Logs, backup and configuration files leakage could be necessary for exploiting web servers.
- Filter input to the server and encode data on the output. It will help to avoid malicious injections.

Technical Details

2.1 Methodology

Our Penetration Testing Methodology grounded on the following guides and standards:

- *Penetration Testing Execution Standard (PTES)*
- *OWASP Top 10 Application Security Risks*
- *OWASP Web Security Testing Guide*
- *Open Source Security Testing Methodology Manual (OSSTMM)*

Penetration Testing Execution Standard (PTES) consists of seven main sections which start from the initial communication and reasoning behind a pentest, through intelligence gathering and threat modeling phases where testers are working behind the scenes to get a better understanding of the tested organization, through vulnerability research, exploitation and post-exploitation, where the technical security expertise of the testers come to play and combine with the business understanding of the engagement, and finally to the reporting, which captures the entire process.

Open Web Application Security Project (OWASP) is an industry initiative for web application security. OWASP has identified the 10 most common attacks that succeed against web applications. Besides, OWASP has created Application Security Verification Standard (ASVS) which helps to identify threats, provides a basis for testing web application technical security controls, and can be used to establish a level of confidence in the security of Web applications.

The Open Source Security Testing Methodology Manual (OSSTMM) is peer-reviewed and maintained by the Institute for Security and Open Methodologies (ISECOM). It has been primarily developed as a security auditing methodology assessing against regulatory and industry requirements. It is not meant to be used as a standalone methodology but rather to serve as a basis for developing one which is tailored towards the required regulations and frameworks.

2.2 Security tools used

- *Manual testing:* Burp Suite Pro [Commercial Edition]
- *Vulnerability scan:* Nessus, OpenVAS, nikto, arachni
- *Network scan:* Nmap, masscan
- *Directory enumeration:* gobuster, dirsearch
- *Injection testing tools:* XSSHunter, SQLmap
- *Encryption:* TestSSL

2.3 Project limitations

The Assessment was conducted against a testing environment with all limitations it provides.

Findings Summary

As an external security company specializing in penetration testing, we have recently conducted a thorough assessment of an application. Our objective was to evaluate the application's security posture and identify any potential vulnerabilities that might compromise the system's integrity.

Our team of certified security professionals employed a combination of manual and automated testing techniques to examine all endpoints within the application. We focused on critical areas such as authentication, authorization, and data validation to ensure that no weak points were left unaddressed.

Throughout our testing process, we adhered to the OWASP Top 10 standards, which represent the most critical security risks to web applications. By simulating real-world attack scenarios, we effectively evaluated the application's resilience against a wide range of potential threats.

Upon completion of our assessment, **we are pleased to report that the application has demonstrated a strong security posture**. The developers have effectively created a secure and robust environment for their users. We can confidently attest to the application's safety, as it has successfully withstood our rigorous penetration testing in compliance with OWASP Top 10 standards.

APPENDIX A - OWASP Web Application Security Top 10

#	Vulnerability	Description	Status
A01	Broken Access Control	Access controls enforce policies so that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure or modification, destruction of data, or performing a business function outside the user's limits.	Safe
A02	Cryptographic Failures	Cryptographic Failures involve protecting data in transit and at rest. This includes passwords, credit card numbers, health records, personal information, and business secrets that require extra protection, especially if that data falls under privacy laws such as GDPR or regulations like PCI Data Security Standard (PCI DSS) for financial data.	Safe

A03	Injection	An application is at risk when user-supplied data is not validated, filtered, or sanitized by the application; dynamic queries or non-parameterized calls without context-aware escaping are used directly in the interpreter; hostile data is used within object-relational mapping (ORM) search parameters to extract additional, sensitive records; or when hostile data is directly used or concatenated.	**Safe**
A04	Insecure Design	According to OWASP, "Secure design is a culture and methodology that constantly evaluates threats and ensures that code is robustly designed and tested to prevent known attack methods. Secure design requires a secure development lifecycle, some form of secure design pattern or paved road component library or tooling, and threat modeling."	**Safe**
A05	Security Misconfiguration	This category includes such things as missing security hardening across any part of the application stack, improperly configured permissions on cloud services, any unnecessary features that are enabled or installed, and unchanged default accounts or passwords. The former category XML External Entities (XXE) is now included in Security Misconfiguration.	**Safe**
A06	Vulnerable and Outdated Components	This category includes any software that is vulnerable, unsupported, or out of date. If you do not know the versions of your components – including all direct and indirect dependencies – or you do not regularly scan and test your components, you are likely at risk.	**Safe**
A07	Identification and Authentication Failures	Security risk occurs when a user's identity, authentication, or session management is not properly handled, allowing attackers to exploit passwords, keys, session tokens, or implementation flaws to assume users' identities temporarily or permanently.	**Safe**

A08	Software and Data Integrity Failures	This includes software updates, critical data, and CI/CD pipelines that are implemented without verification. An example of this includes objects or data encoded or serialized into a structure that an attacker can modify. Another example is an application that relies upon plugins, libraries, or modules from untrusted sources. Insecure CI/CD pipelines that can introduce the potential for unauthorized access, malicious code, or system compromise also fit into this category. Lastly, applications with auto-update functionality, in which updates are downloaded without sufficient integrity verification and applied to a previously trusted application, are considered software and data integrity failures because attackers could infiltrate the supply chain to distribute their own malicious updates.	Safe
A09	Security Logging and Monitoring Failures	This category includes errors in detecting, escalating, and responding to active breaches. Without logging and monitoring, breaches cannot be detected. Examples of insufficient logging, detection, and monitoring include not logging auditable events like logins or failed logins, warnings and errors that generate inadequate or unclear log messages, or logs that are only stored locally. Failures in this category impact visibility, incident alerting, and forensics.	Safe
A10	Server-Side Request Forgery	Server-Side Request Forgery occurs when a web application fetches a remote resource without validating the user-supplied URL. An attacker can coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network ACL. Though SSRF shows a relatively low incidence rate in the data OWASP reviewed, this category was added based on the industry survey results; users are concerned that SSRF attacks are becoming more prevalent and potentially more severe due to increased use of cloud services and the complexity of architectures.	Safe