

Web Penetration Testing Report

[Anonymized]

Please be informed that you currently have an incomplete version of the UnderDefense Web Penetration Testing Report.

[Get full report](#)

Table of contents

Table Of Contents	1
Executive Summary	2
1.1 Project Objectives	3
1.2 Scope & Timeframe	3
1.2.1 Hostnames and IP-addresses	3
1.3 Summary of Findings	5
1.4 Summary of Business Risks	6
1.5 High-Level Recommendations	6
Technical Details	8
2.1 Methodology	11
2.2 Security tools used	11
2.3 Project limitations	11
Findings Details	12
3.1 Critical severity findings	12
3.1.1 Insecure Direct Object Reference in Application Wide	
3.1.2 Web Parameter Tampering	12
3.2.1 SSRF leads to EC2 meta-data disclosure	18
Technical Details	19
2.1 Methodology	9
2.2 Security tools used	9
2.3 Project limitations	9
Findings Details	10
3.1 Critical severity findings	10
3.1.1 Insecure Direct Object Reference in Application Wide	10
3.1.2 Web Parameter Tampering	15
3.1.3 Unauthorized IDOR	21
3.1.4 Unauthorized Stored XSS	25
3.1.5 Insecure Direct Object Reference Leads to Account Takeover	28
3.2 High severity findings	32
3.2.1 Unrestricted File Upload Leads to Stored XSS	32
3.2.2 Stored XSS	38
In blog functionality	38
In the message functionality	40
In client's digital card	41
3.2.3 Cross-site request forgery	44
3.2.4 Insecure Password Reset Functionality Lead To Administrators Account Takeover	48
3.3 Medium severity findings	52
3.3.1 SMS spamming attack via lack of rate limits	52
	55



Table of contents

3.3.2 Weak password policy	55
3.3.2 Weak password policy	55
3.3.3 Unexpirable access link	57
3.3.4 Debug mode enabled	60
3.4 Low severity findings	62
3.4.1 Lack of rate limits	62
3.4.2 User enumeration	65
3.4.3 CSV injection	67
3.4.4 Open redirection	70
3.4.5 Application Logic Bypass	72
3.4.6 CSP not implemented	75
3.4.7 Long Session Timeout	77
3.5 Informational severity findings	78
3.5.1 Server version disclosure	78
3.5.2 Improper error handling	80
APPENDIX A - Security status according to OWASP Top 10	84




Executive Summary

This report presents the results of the “Gray Box” penetration testing for the [CLIENT_NAME] WEB application. The recommendations provided in this report are structured to facilitate remediation of the identified security risks. This document serves as a formal letter of attestation for the recent [CLIENT_NAME] “Gray Box” penetration testing.

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 the [CLIENT_NAME]’s current security as it relates to the [CLIENT_NAME]’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	Security Level	Grade
[CLIENT_NAME] Web Application		Poor

Grade	Security	Criteria Description
	Excellent	The security exceeds “Industry Best Practice” standards. The overall posture was found to be excellent with only a few low-risk findings identified.
	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.
	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
	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.
	Unacceptable	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.

Please be informed that you currently have an incomplete version of the UnderDefense Web Penetration Testing Report. If you're interested in **accessing the comprehensive version**, kindly follow the link provided below

Are you looking for a pentest provider? Contact us!



+1 929 999 5101



underdefense.com

Get full report

1.1 Project Objectives

Our primary goal within this project was to provide the [CLIENT_NAME] 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 [CLIENT_NAME] current security measures with industry best practices
- Providing recommendations that [CLIENT_NAME] 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 [CLIENT_NAME] has been taken in hi to account:

Scope	Confidentiality	Integrity	Availability
All scope objects	High	High	High

1.2 Scope & Timeframe

esting and verification were performed between [DATE]. The scope of this project was limited to the [CLIENT_NAME] application and the specific infrastructure on which the application resides.

We conducted the tests using a production version of the [CLIENT_NAME]. All other applications and servers were out of scope. All testing and verification were conducted from outside of [CLIENT_NAME] offices.

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

1.2.1 Hostnames and IP-addresses

Scope	Description
[WEBSITE]	Administrator web application
[WEBSITE]	Digital card



Scope	Description
[WEBSITE]	Clients web application
[WEBSITE]	API
IP	External IP subnet

1.2.1 User Accounts provided by [CLIENT_NAME]

Asset:	Description
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]

1.2.2 User Accounts provided during testing

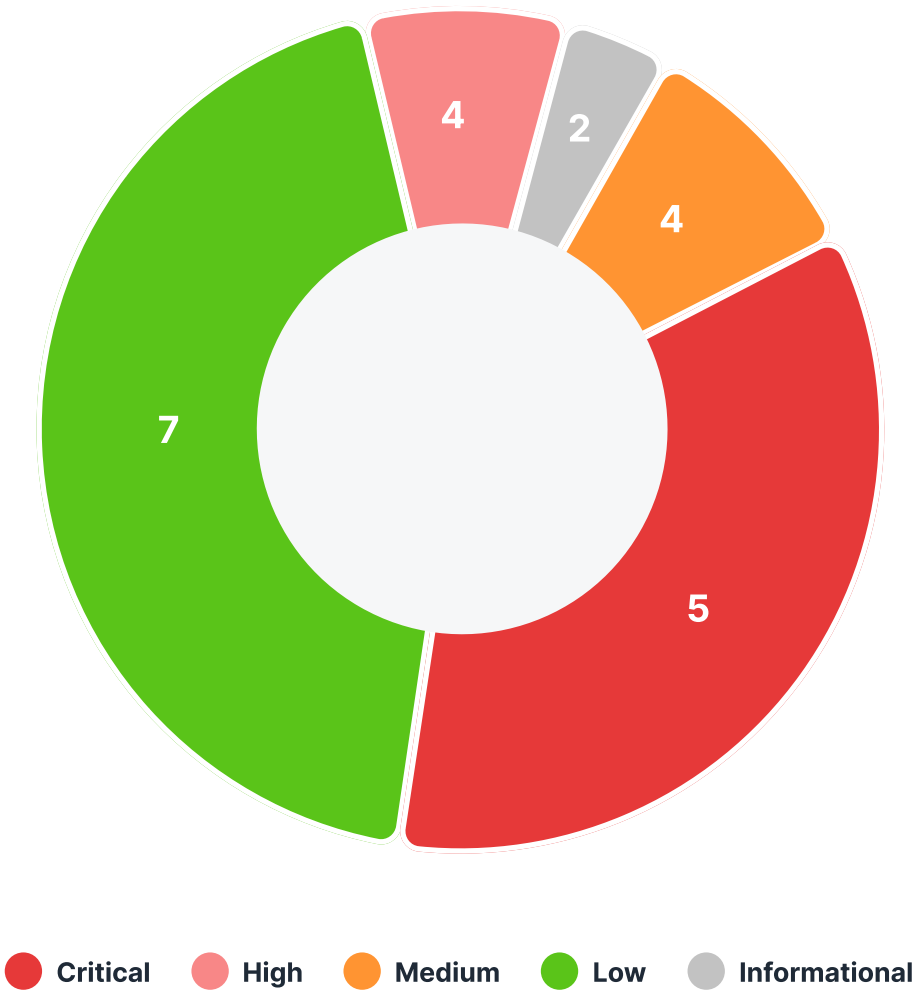
Asset:	Description
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]
[WEBSITE]	[USER_NAME]



1.3 Summary of Findings

Our assessment of the [CLIENT_NAME] application revealed the following vulnerabilities:

Vulnerabilities by severity



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 by types	5	4	4	7	2

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 organization's and users' sensitive information.

1.4 Summary of Business Risks

In the case of [CLIENT_NAME] application issues can lead to:

- Getting access to the admin account by an attacker.
- Customer's sensitive data disclosure.
- Brute forcing users' passwords. Gaining the access to their session after the exploitation of high-level risks.
- Grubbing and exploitation of users' accounts' data.
- Provide additional internal information that can be used by threat actors for successful attacks on a platform.
- Could be chained with other vulnerabilities to increase potential impact.

1.5 High-Level Recommendations

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

- Engage users, especially privileged users, to use 2-factor authentication.
- Improve server and application configuration to meet security best practices.
- Implement strict access control checks.
- When generating CSV output, ensure that formula-sensitive metacharacters are effectively escaped or removed from all data before storage in the resultant CSV.
- Harden your authorization checks on all levels of the access control matrix

- Reconstruct file upload functionality, don't rely on client-side validation only, inspect the content of uploaded files, and enforce a whitelist of accepted, non-executable content types. Additionally, enforce a blacklist of common executable formats, to hinder hybrid file attacks.
- Use strong algorithm for password reset functionality
- Define the default parameters values according to the action that is going to be made by whitelisting them
- Do not provide unauthorized access to accounts registered in the application by error messages, change them to less informative.
- Properly encode user input when accepting it and reflecting back to the user. For that purpose, HTML entities can be used to mitigate XSS attacks.
- Use CSP to enhance immune to XSS attacks

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 Details

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



**Please be informed that this is an
incomplete version of the UnderDefense
Web Penetration Testing Report.**

If you're interested in accessing the comprehensive version, kindly follow the link provided below.

[Get full report](#)

 +1 929 999 5101