Web Application Penetration Test

Prepared for SyncNow

Table of Contents

1. Document Control

Document History

Version	Date	Author	Role	Comments
1.0	12-03-2021	Milan Veljkovic, OSCP, CNSS	Pen Tester	First Release
2.0	17-03-2021			

Document Distribution

Version	Date	Comments
2.0	17-03-2021	Final Release to SyncNow

The following personnel are responsible for handling and distributing this document within the client's organization:

Name	Contact Email	Contact Phone
Aharon Mizrahi	aharon.mizrahi@syncnow.io	**+972 52 4871670**

Copyright

Classification

This document is Client Confidential.

2. Executive Summary

In accordance with instructions provided by Aharon Mizrahi, penetration testing was conducted on the following domain:

- https://syncnowdemo.syncnow.io/

Penetration testing was conducted in period of March 2021. The primary objective of the engagement was to identify significant security vulnerabilities within the systems and help the organization understand the risks associated with these vulnerabilities, to provide solutions to found flaws and retest in order to ensure that flaws are fixed.

In this round of penetration testing, **no flaws were found.**

3. Project Summary

3.1 Introduction

Aharon Mizrahi engaged Milan Veljkovic to conduct web application penetration test of the system within the scope of the engagement to identify any potential security risks and suggest appropriate measures to mitigate the associated risks.

The intent of the engagement was to discover the extent to which an attacker could penetrate and exploit the systems. This test identifies technical vulnerabilities resulting from broken security controls that an attacker will exploit for significant gain.

Testing protocols involve 'black box testing' which means tests are performed without prior information about the target. All information about the target will be obtained at the start of engagement.

3.2 Scope

Penetration testing will be conducted on following target:

- https://syncnowdemo.syncnow.io/

Penetration testing will be conducted during March 2021. Pentester got access to both admin and low level user.

3.3 Summary of Testing Results

This section summarizes all of the vulnerabilities identified by the Pen Tester in the target system.

Operational Risk

The following risk table helps SyncNow to fix issues based on operational risk. This is **not** a list of identified vulnerabilities in the target system, but a list of possible risks for web-based applications:

Reference	Vulnerability	Consequence	Likelihood	Risk
5.1	Stored Cross Site Scripting (XSS)	Major	Likely	**CRITICAL**
5.2	SQL Injection	Major	Likely	**CRITICAL**
5.3	Insufficient File Upload Controls	Major	Possible	**HIGH**
5.4	Insecure Password Policy	Moderate	Likely	**HIGH**
5.5	Logon Available Over HTTP	Moderate	Possible	**MEDIUM**
5.6	Account Enumeration	Moderate	Possible	**MEDIUM**
5.7	Predictable Session Token	Moderate	Possible	**MEDIUM**
5.8	Insecure Password Storage or Transmission	Major	Unlikely	**MEDIUM**
5.9	Form Auto Complete Enabled	Moderate	Possible	**MEDIUM**
5.10	Directory Listing	Minor	Possible	**LOW**
5.11	Insufficient Session Termination	Moderate	Unlikely	**LOW**
5.12	Cross Domain Script Include	Moderate	Unlikely	**LOW**
5.13	Insufficient Input Validation	Minor	Possible	**LOW**
5.14	File System Path Disclosure	Minor	Unlikely	**LOW**

4. Web Application Issue Categories Summary

Introduction

This section summarizes the Issue Categories that have been tested and the corresponding results. It also highlights Issue Categories that fell out of the scope of the engagement or were not applicable. The following icons denote the Issue Category results:

✓ **Yes** – Tests were executed within the Issue Category and all tests passed.

✗ **Yes** – Tests were executed within the Issue Category and identified security weakness via one or more of the tests. The organization should review and mitigate the associated risk.

! **No** – Tests were not executed within the Issue Category, as it was not relevant to the system, was out of scope of the engagement, or alternatively, test was unable to be completed.

Test ID	Test Name	OWASP Top 10	Result
1	Authentication	A2: Broken Authentication and Session Management	✓
2	Authorization	A4: Insecure Direct Object Reference A7: Missing Function Level Access Control	✓
3	Session Management	A2: Broken Authentication and Session Management A8: Cross-Site Request Forgery (CSRF)	✓
4	Business Logic		✓
5	Data Validation	A1: Injection A3: Cross-Site Scripting (XSS) A10: Unvalidated Redirects and Forwards	✓
6	Data Privacy	A6: Sensitive Data Exposure	✓
7	Availability		✓
8	Auditing and Logging		!
9	Exception Handling	A5: Security Misconfiguration	✓
10	Information Leakage	A5: Security Misconfiguration	✓
11	Configuration Management	A5: Security Misconfiguration A9: Using Components with Known Vulnerabilities	✓

Test ID	ID	SANS 25 top vulnerabilities	Result
1	https://cwe.mitre.org/data/definitions/119.html	Improper restriction of operations withing the bound of a memory buffer	✓
2	https://cwe.mitre.org/data/definitions/79.html	Improper neutralization of input during web page generation("Cross-site scripting")	✓
3	https://cwe.mitre.org/data/definitions/20.html	Improper input validation	✓
4	https://cwe.mitre.org/data/definitions/200.html	Information exposure	✓
5	https://cwe.mitre.org/data/definitions/125.html	Out-of-bounds read	✓
6	https://cwe.mitre.org/data/definitions/89.html	Improper neutralization of special elements used in an SQL command ("SQL injection")	✓
7	https://cwe.mitre.org/data/definitions/416.html	Use after free	!
8	https://cwe.mitre.org/data/definitions/190.html	Integer overflow of wraparound	✓
9	https://cwe.mitre.org/data/definitions/352.html	Cross-Site request forgery (CSRF)	✓
10	https://cwe.mitre.org/data/definitions/22.html	Improper limitation of a pathname to a restricted directory ("Path traversal")	✓
11	https://cwe.mitre.org/data/definitions/78.html	Improper neutralization of special elements used in an OS command ("OS command injection")	✓
12	https://cwe.mitre.org/data/definitions/787.html	Out-of-bounds write	✓
13	https://cwe.mitre.org/data/definitions/287.html	Improper authentication	✓
14	https://cwe.mitre.org/data/definitions/476.html	NULL Pointer dereference	✓
15	https://cwe.mitre.org/data/definitions/732.html	Incorrect permission assignment for critical resource	✓
16	https://cwe.mitre.org/data/definitions/434.html	Unrestricted upload of a file with dangerous type	✓
17	https://cwe.mitre.org/data/definitions/611.html	Improper restriction of XML external entity reference	✓
18	https://cwe.mitre.org/data/definitions/94.html	Improper control of generation of code ("Code injection")	✓

19	https://cwe.mitre.org/data/definitions/798.html	Use of hard-coded credentials	✓
20	https://cwe.mitre.org/data/definitions/400.html	Uncotrolled resource consumption	✓
21	https://cwe.mitre.org/data/definitions/772.html	Missing release of resource after effective lifetime	✓
22	https://cwe.mitre.org/data/definitions/426.html	Untrusted search path	✓
23	https://cwe.mitre.org/data/definitions/502.html	Deserialization of untrusted data	✓
24	https://cwe.mitre.org/data/definitions/269.html	Improper privilege management	✓
25	https://cwe.mitre.org/data/definitions/295.html	Improper certificate validation	✓

5. Web Application Vulnerabilities

No flaws were found in this web application.

6. Appendix: Vulnerability Risk Evaluation Methodology

Pen testing discovers vulnerabilities and provides a lot of different information about the vulnerabilities described in this report. It provides a description, a technical risk assessment, a factual write up of disclosures, exploits, outcomes, and recommended actions to mitigate the risk.

Pen testing recommendations come from years of experience and industry "Best Practice" standards. The client is free to choose another approach that better suits the client's environment or skills as long as the solution eliminates the possibility of reproducing the exploit.

Pen testing conducts a qualitative risk-analysis on each vulnerability in this report. The sections below discuss how we have arrived at the final risk assessment values.

7 Overall Risk Evaluation Methodology

We take a popular approach to evaluating and expressing the technical risk associated with each vulnerability. The table below depicts a standard risk table that is includes in the "Risk Table" section of the report:

Reference	Vulnerability	Likelihood	Consequence	Technical Risk	Technical Impact

The overall risk is expressed within the "Technical Risk" column of the risk table. This represents the risk that the vulnerability exposes to the underlying systems that support the business. The methodology uses a popular qualitative risk model to describe the risk.

The sections below discuss how to arrive at the final values for the technical risk.

Overall Technical Risk Calculation

Pen testing determines technical risk using a qualitative approach. It is roughly equivalent to its quantitative counterpart. To calculate quantitative risk, take the dot product of the numeric values of impact and likelihood:

$$\text{Technical Risk} = \text{Impact} \cdot \text{Likelihood}$$

We then use the qualitative equivalent to express risk through a risk table:

		Consequence				
Likelihood		Insignificant	Minor	Moderate	Major	Critical
	Almost certain	MEDIUM	HIGH	EXTREME	EXTREME	EXTREME
	Likely	LOW	MEDIUM	HIGH	EXTREME	EXTREME
	Possible	LOW	LOW	MEDIUM	HIGH	EXTREME
	Unlikely	LOW	LOW	LOW	MEDIUM	HIGH
	Rare	LOW	LOW	LOW	LOW	MEDIUM

Each final risk value is a cell in the above table. A technical risk value is determined through the intersection of a likelihood value and a technical impact value.

The "Likelihood" column of the risk table represents the likelihood that an attacker will exploit the vulnerability. The "Consequence" column of the risk table represents the impact that may result from the attack. The "Technical Impact" column of the risk table provides a brief custom description of the nature of the vulnerability.

Subsequent sections below outline each of these dimensions to risk.

Consequence Values

Within each finding, we estimate the technical impact from that vulnerability on the organisation. An attack affects three valuable assets to the organisation:

- **User** – If the attacker exploits the vulnerability, the attacker may compromise the users' sense of trust, safety, or security with the organisation. This may result in reputational damage to the organisation.
- **Data** – If the attacker exploits the vulnerability, the attacker may compromise the organisation's information assets. This may result in violations of data confidentiality, integrity, or availability.
- **Environment** – If the attacker exploits the vulnerability, the attacker may compromise the organisation's infrastructure. This may result in unauthorised changes to the underlying IT infrastructure. These changes may result in disruption of business activity or denial of service.

The final value in the "Consequence" row of the risk table represents the maximum impact possible when considering these three different impacts to the organisation, as illustrated below:

Description	Consequence of Occurrence
Critical	The consequences may gravely threaten major organisation objectives. Financial implications may have extreme consequences for the organisation.
Major	The consequences may threaten the continued effective functioning of the organisation. Financial implications may have very high consequences for the organisation.
Moderate	The consequences should not threaten the program /project or department but may mean that the program or project could be subject to significant review and/or operational change. Financial implications may have medium consequences for the organisation.
Minor	The consequences should only threaten the efficiency or effectiveness of some aspects of the program/project or department. However, the organisation could deal with this internally. Any financial implication would be of low consequence.
Insignificant	The organisation can easily deal with the consequences by routine operations. Any financial implication would be of negligible impact.

Likelihood Calculation

Within each finding, we provide an estimation of the likelihood that an attacker will exploit the vulnerability. A number of different factors influence the likelihood-of-attack:

- **Ease of exploitability** – A vulnerability that is easy to exploit represents "low hanging fruit."
- **Skill Level** – How much skill is required to exploit the vulnerability?
- **Motivation** – Is there a strong enough motivation to exploit the vulnerability?
- **Opportunity** – Does the attacker require special tools to conduct the attack?
- **Popularity** – Is the vulnerability a well-known type within the security community?
- **Intrusion Detection** – Will the attacker think that the organisation will be actively looking for any type of activity associated with the exploitation of the vulnerability?

The following likelihood-values table shows risk measures and their frequency of occurring:

Description	Likelihood of Occurrence
Almost certain	Is expected to occur in most circumstances
Likely	Will probably occur in most circumstances
Possible	Could occur at some time
Unlikely	Occurrance is considered low
Rare	May occur only in exceptional circumstances

8 Appendix: Application Security Testing Methodology

This section covers the systematic approach followed while performing a security assessment.

During an engagement, we conduct a number of tests that include both tool-based analysis using common assessment tools, as well as experienced specialists performing manual testing techniques against target systems.

During the assessment, the team performs automated testing with an objective to identify known common attacks against the systems (that typically are extremely easy to locate) whilst hands-on type testing discovers more obscure (and difficult to exploit) vulnerabilities and validate any findings from the tools results were applicable.

Where possible we allocate as much time as possible to each phase, however it is not always possible to identify all vulnerabilities in the limited time allocated against each phase of the project.

8.1 Application Penetration Testing

Application testing begins with an initial "passive reconnaissance phase" where application and host-infrastructure is analysed without launching any attacks to help identify attack vectors. In the initial phase (if possible) commercial and open source automated testing tools are used against the target applications.

Once the team has performed an initial analysis, consultants use a number of manual attack techniques and tools (provided below) in performing the assessment against the applications. These manual techniques focus on identifying the vulnerabilities listed in the section 3 to section 12 (covers OWASP Top 10, CWE Top 25 and SANS Top 20). However, if the consultant believes additional attack vectors would be successful, they perform these too.

8.2 Testing Approach

The pen testing security testing methodology includes structured review processes based on recognized "best-in-class" practices as defined by such organizations as NIST, ISECOM (OSSTMM) and OWASP.

This methodology suggests the following high-level phases for every security-testing project:

- Preparation and planning
- Passive and active reconnaissance
- Vulnerability detection and verification (Exploitation)
- Analysis and Reporting
- Cleaning up (Restoring system state)

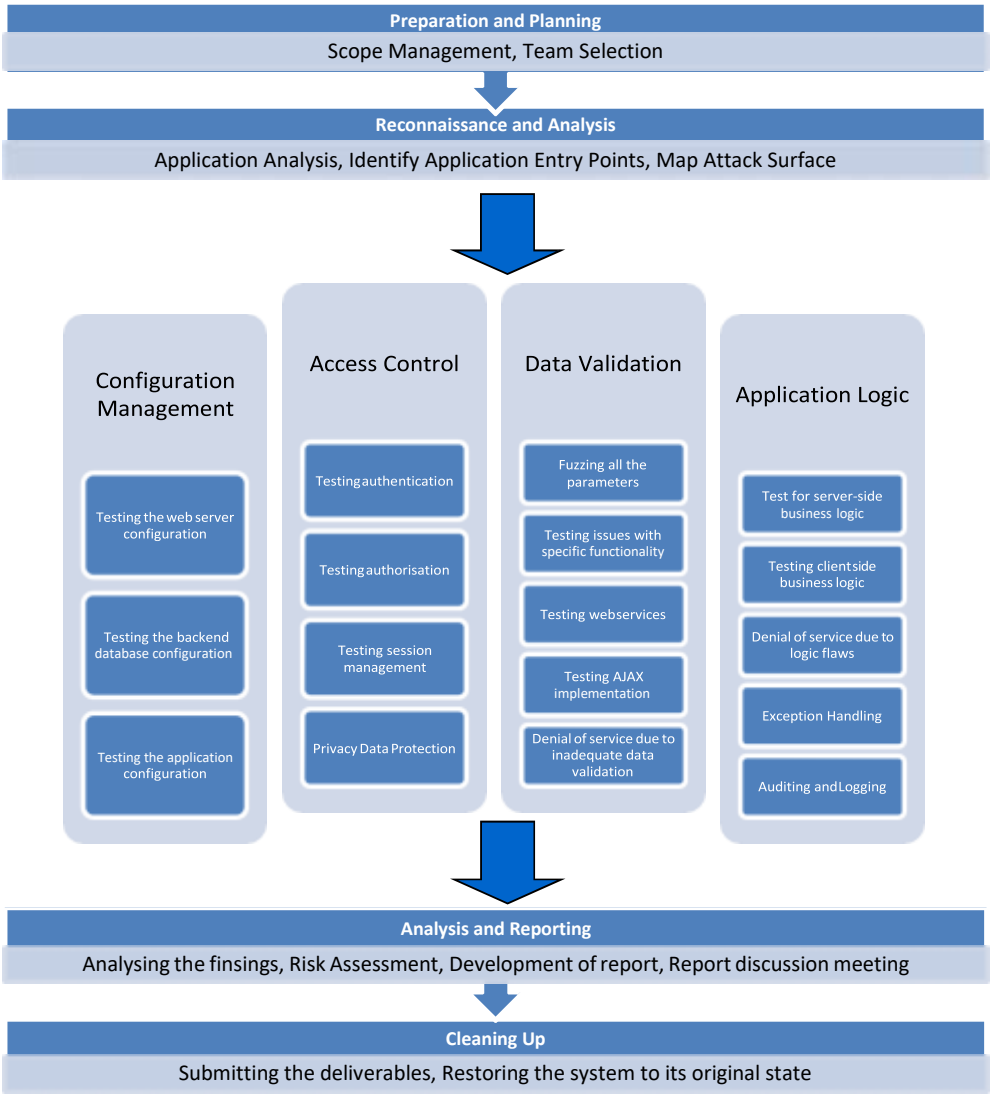

Figure 20. Pen Tester Application Testing Methodology

8.3 Information Gathering (Passive and Active Reconnaissance)

The actual penetration test begins with a comprehensive evaluation of the target application. In this phase, the team tries to gather as much information as possible about the target system. This information is gathered through active (probing the application directly) as well as passive (through third party systems e.g. Google, Blogs etc.) means. The team tries to collect the following information:

- Network architecture or application design
- Open/Closed ports/services
- Operating system information and service enumeration
- Any vulnerability associated with the hosting environment or OS being used by the organization

- Applications and attack vectors associated with application services

8.4 Vulnerability Detection and Verification

After reconnaissance is complete, the security consultant utilizes comprehensive set of tools and manual techniques to identify and exploit the target systems. Based on the findings from the scan and previous phases, an attacker executes specific attacks against the application. These attacks test the target systems for the following:

- Information Leakage
- Configuration Management Testing
- Authentication
- Authorisation
- Business Logic
- Data Validation
- Data Protection
- Denial of Service
- Auditing and Logging
- Exception Handling
- Web Services specific vulnerabilities
- AJAX specific vulnerabilities

The aim of this phase is to gain access to the key systems or services such as domain controllers, email platforms, critical databases, or misuse of application services. Section 9 outlines the vulnerabilities.

8.5 Result Analysis and Reporting

This phase involves analysing all the attacks performed on the target systems and compiling a report based on the findings. All the successful attacks conducted are included in the report with recommendation/advice on fixing the issue.

Traditionally pen testing consultants will capture as much information as possible during the testing program such as examples; screen captures and attack techniques to allow for ease of understanding in the reports.

8.6 Cleaning Up

This is the last phase of the project where the team completes the analysis, reviews the results of the assessment and ensures the collection of results.

We submit a draft report to the client detailing the results of the assessment. From here, the client

provides comments and the consultants prepare a final version for the client.

Before the client receives the final report, the team destroys any evidence of their attacks. The team removes all information and then gives the report to the client for storage.

9 Appendix: Web Penetration Testing Tools

Burp Suite

Burp Suite is an integrated platform for attacking web applications. It contains the entire burp toolset (proxy, spider, intruder and repeater) with numerous interfaces between them designed to facilitate and speed up the process of attacking a web application. All plug-ins share the same robust framework for handling HTTP requests, authentication, downstream proxies, logging, alerting and extensibility.

http://portswigger.net/burp/

Google Search Engine

Google is widely recognised as the "world's best search engine" because it is fast, accurate and easy to use. It allows users to search for information relating to almost anything available on the Internet.

http://www.google.com

Internet Explorer and Firefox Plugins

Pen Tester uses a wide range of security plugins. These allow the team to modify requests, inspect web page internal structure and perform a range of attacks within the browser itself. The list of plugins includes:

7. Access Control ME – access control testing
8. Advanced Dork – resurrect pages from search engine caches
9. Cache Viewer – inspect cached elements
10. Default User Agent – change user agents on the fly
11. Fiddler – IE toolbar similar to Web Developer Toolbar
12. Firebug – Firefox DOM inspector and script debugger and profiler
13. Grease Monkey – run arbitrary scripts against a page
14. GWT Decode bookmarklet – Decode GWT endpoints
15. Hackbar – Encoder and other utilities
16. Java Console – Useful for tampering with applets
17. JavaScript Deobfuscator – Unpack packed sources

18. Page Info – extended information about pages
19. REST Client – tamper with REST APIs
20. RESTtest – test REST APIs
21. Show IP – show the current host's IP address
22. SQL Inject ME – SQL injection tool
23. Tamper Data – proxyless tampering tool
24. XSS ME – XSS testing tool
25. Web Developer Toolbar – extremely useful utility plugin

Metasploit Framework

The Metasploit Framework is an advanced open-source platform for developing, testing, and using exploit code. This project is a powerful tool for penetration testing, exploit development, and vulnerability research.

http://www.metasploit.com

Nmap

Nmap is an open source utility for network exploration or security auditing. It rapidly scans large networks, although it works fine against single hosts. Nmap uses raw IP packets in ways to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and OS version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.

http://insecure.org

OpenSSL

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general-purpose cryptography library.

http://www.openssl.org

OWASP DirBuster

DirBuster is a powerful directory and file brute force tool. DirBuster comes with several lists of approximately 2.3 million common files and directories, and can brute force any file or directory given enough time.

http://www.owasp.org/index.php/Category:OWASP_DirBuster_Project

sqlmap

This tool automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It allows for timely SQL Injection exploitation to access the backend database and the underlying Operating System.

http://sqlmap.org/

Wikto

Wikto is a Web server scanner that performs comprehensive tests against web servers for multiple items. This includes over 2200 potentially dangerous files/CGIs, versions on over 140 servers, and problems on over 210 servers. The tool will automatically update items and plugins if the user desires.

http://www.sensepost.com

OWASP Zap

OWASP ZAP is an open-source web application security scanner. It is intended to be used by both those new to application security as well as professional penetration testers. It is one of the most active Open Web Application Security Project projects and has been given Flagship status.

Nessus

Nessus is a proprietary vulnerability scanner developed by Tenable, Inc.