Offensive Security   
OSWA Exam Report

your@mail.com  
OSID: XXXXXX

Author:

Date: July 12, 2022

Auteur:

Datum : 10 februari 2021

# OSWA Exam Report

|  |  |
| --- | --- |
| Document properties | |
| Details | |
| Author | Your Name |
| OSID | XXXXXX |
| Mail | your@mail.com |
| Document number | OSWA-OS-012345 |
| Date | July 1, 2022 |
| Status | Final |

Table 1: Details of the report.

|  |  |  |
| --- | --- | --- |
| Version history | | |
| Date | Version | Description |
| July 1, 2022 | 0.1 | First version |
| July 1, 2022 | 1.0 | Final |

Table 2: Version history of the report.

# Content

[OSWA Exam Report 1](#_Toc108519540)

[Content 3](#_Toc108519541)

[Management Summary 4](#_Toc108519542)

[Significant Results 4](#_Toc108519543)

[Recommendations 4](#_Toc108519544)

[1 Methodologies 5](#_Toc108519545)

[1.1 Information Gathering 5](#_Toc108519546)

[1.2 Service and Application Enumeration 5](#_Toc108519547)

[1.3 Penetration 5](#_Toc108519548)

[2 Penetration 6](#_Toc108519549)

[2.1 Target #1 – 192.168.XX.XX 6](#_Toc108519550)

[2.2 Target #2 – 192.168.XX.XX 8](#_Toc108519551)

[2.3 Target #3 – 192.168.XX.XX 10](#_Toc108519552)

[2.4 Target #4 – 192.168.XX.XX 11](#_Toc108519553)

[2.5 Target #5 – 192.168.XX.XX 14](#_Toc108519554)

[3 Appendix 15](#_Toc108519555)

# Management Summary

I was tasked with performing an internal penetration test towards Offensive Security Exam. An internal penetration test is a dedicated attack against internally hosted applications. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security’s internal exam systems. My overall objective was to evaluate the web applications and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security’s web applications. When performing the attacks, I was able to gain access to multiple machines, primarily due to insecure code and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited, and access granted.

## Significant Results

The exploited applications as well as a brief description on how access was obtained are listed below.

|  |  |  |
| --- | --- | --- |
| IP | Hostname | Exploit |
| 192.168.XX.XX | example.com | SQLi |
| 192.168.XX.XX | example.com | SSTI |
| 192.168.XX.XX | example.com | XSS |
| 192.168.XX.XX | example.com | Command Injection |
| 192.168.XX.XX | example.com | SSRF |

*Table 3: The systems in scope that were exploited.*

## Recommendations

I recommend resolving the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these applications in the future. One thing to remember is that these applications require maintenance, to protect for additional vulnerabilities that are discovered at a later date.

# Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

## Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting exam applications. The specific IP addresses were:

|  |
| --- |
| IP |
| 192.168.XX.XX |
| 192.168.XX.XX |
| 192.168.XX.XX |
| 192.168.XX.XX |
| 192.168.XX.XX |

Table 4: Details of the application scope.

## Service and Application Enumeration

The system and application enumeration portion of a penetration test focuses on gathering information about what services are alive on a system and what directories and files are publicly accessible from the application. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications and services are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

## Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, Tester was able to successfully gain access to X out of the X systems.

# Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to X out of the X systems.

## Target #1 – 192.168.XX.XX

### Flags

|  |  |
| --- | --- |
| Flags | Value |
| Local.txt |  |
| Proof.txt |  |

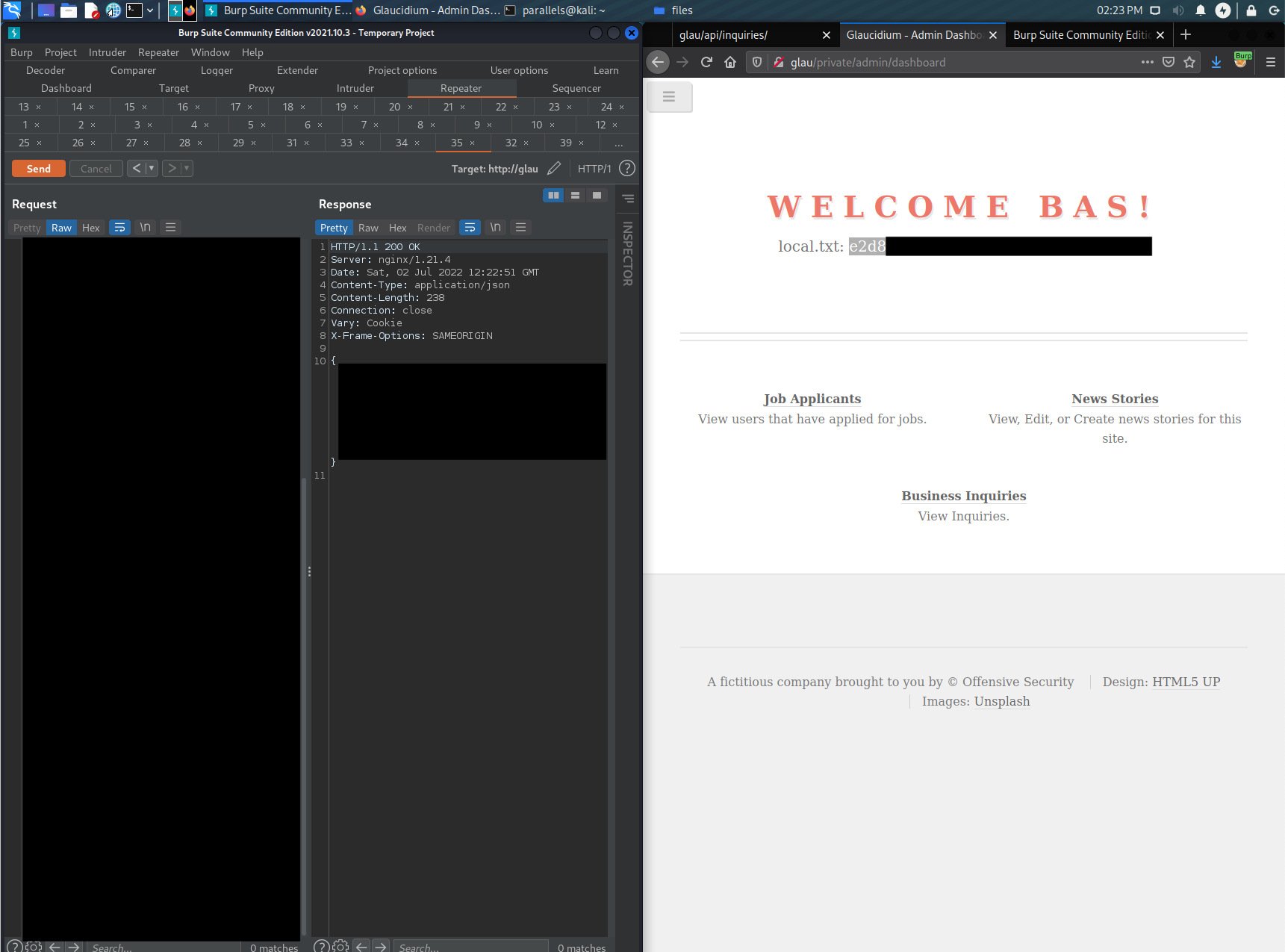


Figure 1: Local.txt screenshot



Figure 2: Proof.txt screenshot

### Application Enumeration

#### Hostname

example.com

#### Port Scan Results

|  |  |
| --- | --- |
| IP | Ports Open |
| 192.168.xx.xx | TCP: |

#### Subdomains

test.example.com

#### Directories and Files

/test.php

### Vulnerability: XSS – Initial Access

|  |  |
| --- | --- |
| Vulnerability ID | VULN-001 |
| Severity | Critical |
| Affected resources | <http://example.com>/status.php |

#### Vulnerability Explanation

Explanation of the vulnerability.

#### Evidence

The following steps can be taken to reproduce this finding:

1. Login to the application at of <http://example.com>
2. Browse to <http://example.com/status.php>
3. Post a status message with the following contents

<script>alert(‘XSS’)</script>

1. Reload the status page

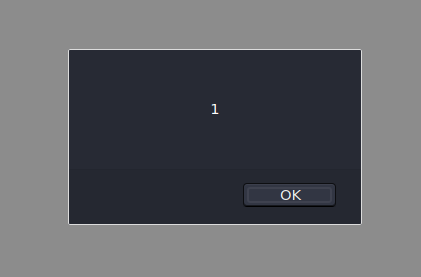


Figure 3: JavaScript payload is executed.

#### Remediation

How to fix.

## Target #2 – 192.168.XX.XX

### Flags

|  |  |
| --- | --- |
| Flags | Value |
| Local.txt |  |
| Proof.txt |  |

Graphical user interface

Description automatically generated

Figure 1: Local.txt screenshot



Figure 2: Proof.txt screenshot

### Application Enumeration

#### Hostname

example.com

#### Port Scan Results

|  |  |
| --- | --- |
| IP | Ports Open |
| 192.168.xx.xx | TCP: |

#### Subdomains

test.example.com

#### Directories and Files

/test.php

### Vulnerability: XSS – Initial Access

|  |  |
| --- | --- |
| Vulnerability ID | VULN-002 |
| Severity | Critical |
| Affected resources | <http://example.com>/status.php |

#### Vulnerability Explanation

Explanation of the vulnerability.

#### Evidence

The following steps can be taken to reproduce this finding:

1. Login to the application at of <http://example.com>
2. Browse to <http://example.com/status.php>
3. Post a status message with the following contents

<script>alert(‘XSS’)</script>

1. Reload the status page

Graphical user interface

Description automatically generated

Figure 3: JavaScript payload is executed.

#### Remediation

How to fix.

## Target #3 – 192.168.XX.XX

### Flags

|  |  |
| --- | --- |
| Flags | Value |
| Local.txt |  |
| Proof.txt |  |

Graphical user interface

Description automatically generated

Figure 1: Local.txt screenshot



Figure 2: Proof.txt screenshot

### Application Enumeration

#### Hostname

example.com

#### Port Scan Results

|  |  |
| --- | --- |
| IP | Ports Open |
| 192.168.xx.xx | TCP: |

#### Subdomains

test.example.com

#### Directories and Files

/test.php

### Vulnerability: XSS – Initial Access

|  |  |
| --- | --- |
| Vulnerability ID | VULN-003 |
| Severity | Critical |
| Affected resources | <http://example.com>/status.php |

#### Vulnerability Explanation

Explanation of the vulnerability.

#### Evidence

The following steps can be taken to reproduce this finding:

1. Login to the application at of <http://example.com>
2. Browse to <http://example.com/status.php>
3. Post a status message with the following contents

<script>alert(‘XSS’)</script>

1. Reload the status page

Graphical user interface

Description automatically generated

Figure 3: JavaScript payload is executed.

#### Remediation

How to fix.

## Target #4 – 192.168.XX.XX

### Flags

|  |  |
| --- | --- |
| Flags | Value |
| Local.txt |  |
| Proof.txt |  |

Graphical user interface

Description automatically generated

Figure 1: Local.txt screenshot



Figure 2: Proof.txt screenshot

### Application Enumeration

#### Hostname

example.com

#### Port Scan Results

|  |  |
| --- | --- |
| IP | Ports Open |
| 192.168.xx.xx | TCP: |

#### Subdomains

test.example.com

#### Directories and Files

/test.php

### Vulnerability: XSS – Initial Access

|  |  |
| --- | --- |
| Vulnerability ID | VULN-004 |
| Severity | Critical |
| Affected resources | <http://example.com>/status.php |

#### Vulnerability Explanation

Explanation of the vulnerability.

#### Evidence

The following steps can be taken to reproduce this finding:

1. Login to the application at of <http://example.com>
2. Browse to <http://example.com/status.php>
3. Post a status message with the following contents

<script>alert(‘XSS’)</script>

1. Reload the status page

Graphical user interface

Description automatically generated

Figure 3: JavaScript payload is executed.

#### Remediation

How to fix.

## Target #5 – 192.168.XX.XX

### Flags

|  |  |
| --- | --- |
| Flags | Value |
| Local.txt |  |
| Proof.txt |  |

Graphical user interface

Description automatically generated

Figure 1: Local.txt screenshot



Figure 2: Proof.txt screenshot

### Application Enumeration

#### Hostname

example.com

#### Port Scan Results

|  |  |
| --- | --- |
| IP | Ports Open |
| 192.168.xx.xx | TCP: |

#### Subdomains

test.example.com

#### Directories and Files

/test.php

### Vulnerability: XSS – Initial Access

|  |  |
| --- | --- |
| Vulnerability ID | VULN-001 |
| Severity | Critical |
| Affected resources | <http://example.com>/status.php |

#### Vulnerability Explanation

Explanation of the vulnerability.

#### Evidence

The following steps can be taken to reproduce this finding:

1. Login to the application at of <http://example.com>
2. Browse to <http://example.com/status.php>
3. Post a status message with the following contents

<script>alert(‘XSS’)</script>

1. Reload the status page

Graphical user interface

Description automatically generated

Figure 3: JavaScript payload is executed.

#### Remediation

How to fix.

# Appendix

This section is placed for any additional items that were not mentioned in the overall report.