Web Application Security Assessment Report

Executive Summary

Overview

I was engaged to conduct a Web application Security Assessment on *.terahost.exam. The purpose of the engagement was to utilize active exploitation techniques in order to evaluate the security of the application against best practice criteria and to validate its security mechanisms and identify application level vulnerabilities.

A Web Application Security Assessment provides stakeholders with insight into the resilience of an application to withstand attack from unauthorized users and the potential for valid users to abuse their privileges and access. The assessment evaluates the security of the application against best practice criteria to validate security mechanisms and identify application level vulnerabilities.

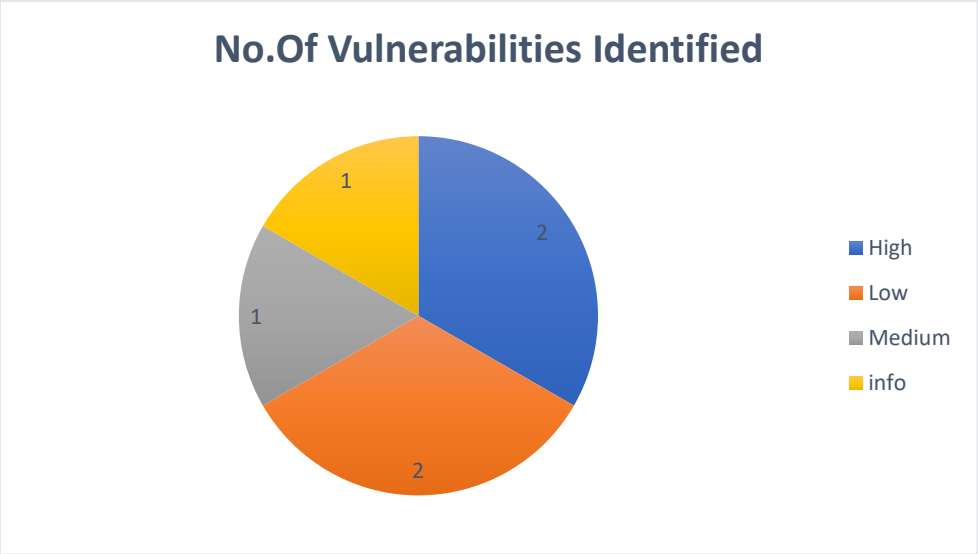
This report details the scope of testing conducted, all significant findings along with detailed remedial advice. The summary below provides a non-technical audience with a summary of the key findings and relates these back to business impacts. Section two of this report relates the key findings. Section three of this report provides detailed narration and individual vulnerability findings that are aimed at a technical audience.

This document summarizes the findings, analysis and recommendations from the assessment, which was conducted on the lab environment.

Summary of Findings

The graph below shows a summary of the number of vulnerabilities found for each impact level for the Web Application Security Assessment.

Severity	Vulnerability count
High	02
Medium	01
Low	02
Informational	01



2. Introduction

2.1 Overview

This report documents the findings for the Web Application Security Assessment of the terahost application.

The purpose of the engagement was to utilize exploitation techniques in order to identify and validate potential vulnerabilities across all systems within scope.

2.2 Scope

Activity performed a Web Application Security Assessment of the *.terahost.exam application.

2.3 Project Team

The engagement began on 26th November 2021 and involved contributions from the following members:

Role	Name
Security Analyst	Anil Arelli

3. Key Findings

3.1 Introduction

This section outlines a summary of the key issues identified during the course of the assessment. A qualitative impact factor (High, Medium, or Low) has been assigned to each vulnerability identified. However, all of the detailed findings in section four of this report should be reviewed and the recommended corrective action implemented where appropriate.

3.2 Key Issues

Following are the statistics according to the severity:

Sr. No.	Vulnerability Name	Severity	Control Area	Owasp Category
1.	Cross site scripting	**Medium**	Cross-Site Scripting (XSS)	A7
2.	Privilege escalation	HIGH	Broken access control	A5
3	Sql injection	High	Injection	A9
4	Click jacking	Low	Security Misconfigurations	A6
5	PHP info page	Low	Security Misconfigurations	A6
6	Missing security Headers	Info	Security Misconfigurations	A6

4. Detailed Findings

4.1 cross site scripting

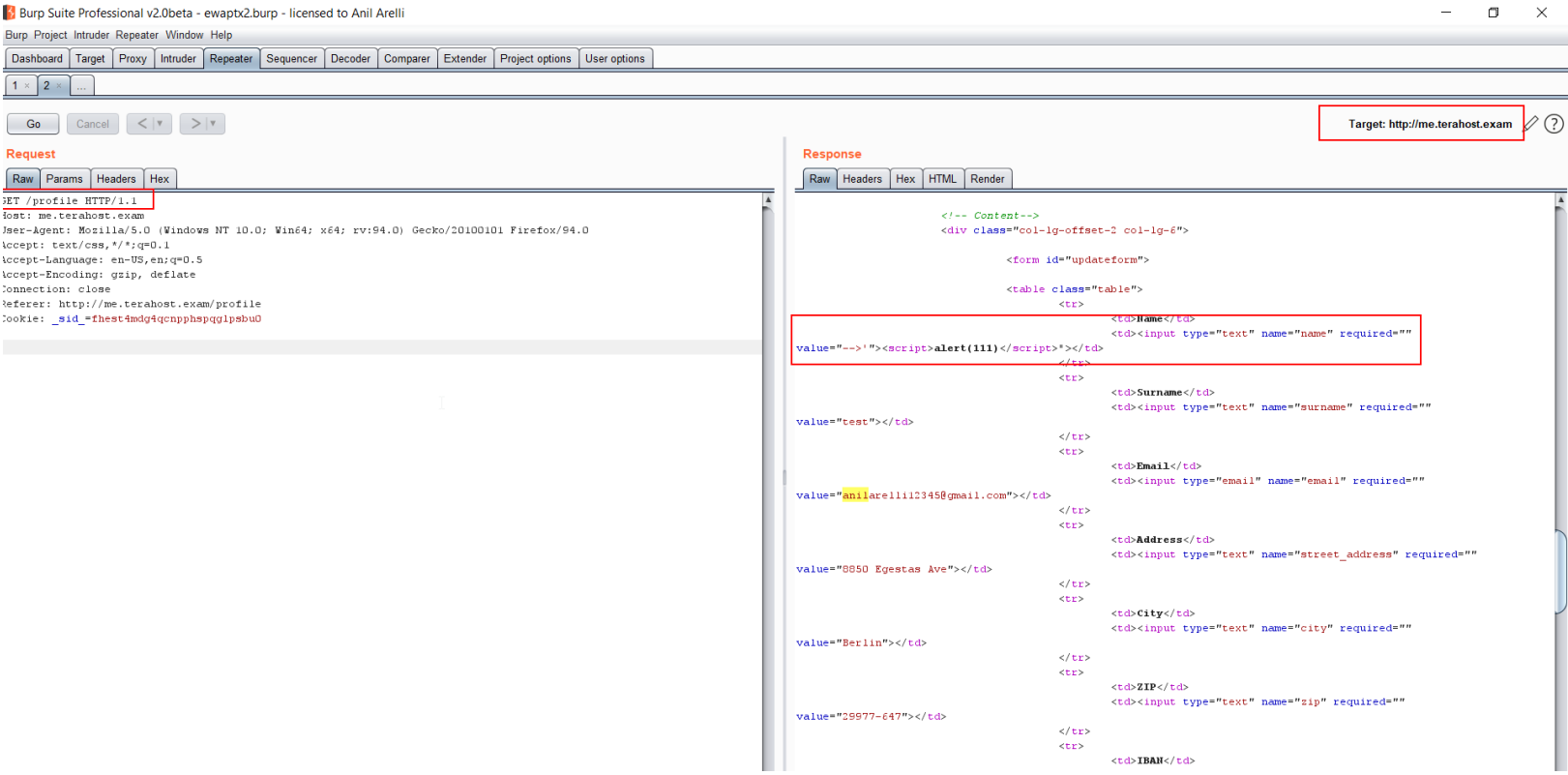
Risk Rating: Medium

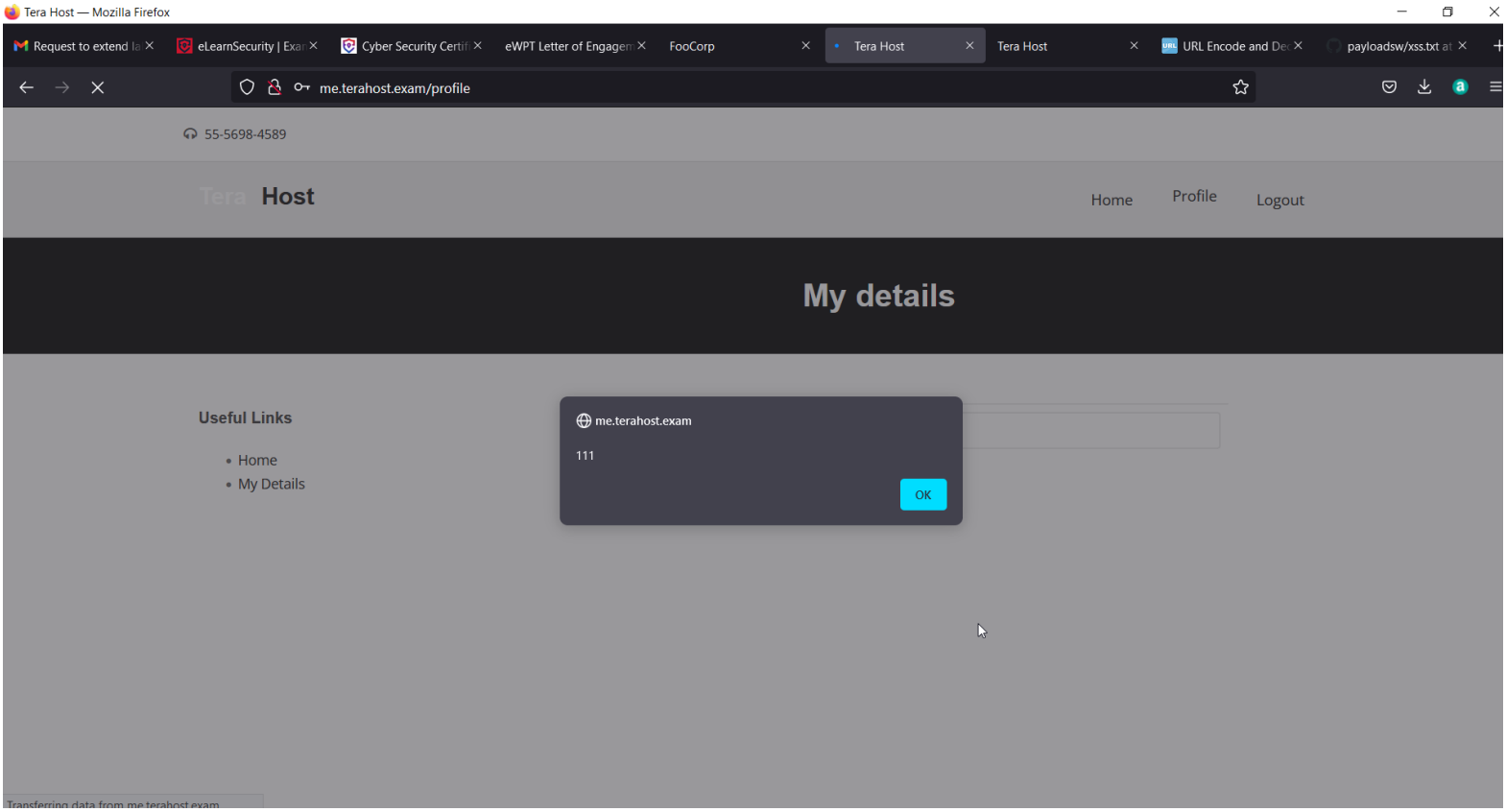
Analysis: Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Stored attacks are those where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc.

POC1 :-

XSS in firstname parameter in me.terahost.exam

Open the /profile page and save the firstname with xss payload, Now load the profile page XSS gets triggered as show below .





XSS in surname parameter

Open the /profile page and save the surname with xss payload, Now load the profile page XSS gets triggered as show below .

Burp Suite Professional v2.0beta - ewaptx2.burp - licensed to Anil Arelli

Burp Project Intruder Repeater Window Help

Dashboard | Target | Proxy | Intruder | Repeater | Sequencer | Decoder | Comparer | Extender | Project options | User options

1 × 2 × ...

Go | Cancel | < | > | Target: http://me.terahost.exam

Request

Raw | Params | Headers | Hex

```
GET /profile HTTP/1.1
Host: me.terahost.exam
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: text/css,*/*;q=0.1
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://me.terahost.exam/profile
Cookie: _sid_=fhest4mdg4qcnpphspqglpsbu0
```

Response

Raw | Headers | Hex | HTML | Render

```
                <!-- End Sidebars-->

                <!-- Content-->
                <div class="col-lg-offset-2 col-lg-6">

                        <form id="updateform">

                        <table class="table">
                                <tr>
                                        <td>Name</td>
                                        <td><input type="text" name="name" required="" value="test"></td>
                                </tr>
                                <tr>
                                        <td>Surname</td>
                                        <td><input type="text" name="surname" required=""
value="-->'"><script>alert(111)</script>*></td>
                                </tr>
                                <tr>
                                        <td>Email</td>
                                        <td><input type="email" name="email" required=""
value="anilarelli12345@gmail.com"></td>
                                </tr>
                                <tr>
                                        <td>Address</td>
                                        <td><input type="text" name="street_address" required=""
value="8850 Egestas Ave"></td>
                                </tr>
                                <tr>
                                        <td>City</td>
                                        <td><input type="text" name="city" required=""
value="Berlin"></td>
                                </tr>
                                <tr>
                                        <td>ZIP</td>
                                        <td><input type="text" name="zip" required=""
value="29977-647"></td>
                                </tr>
```

me.terahost.exam/profile

55-5698-4589

Tera Host

Home | Profile | Logout

My details

Useful Links

- Home
- My Details

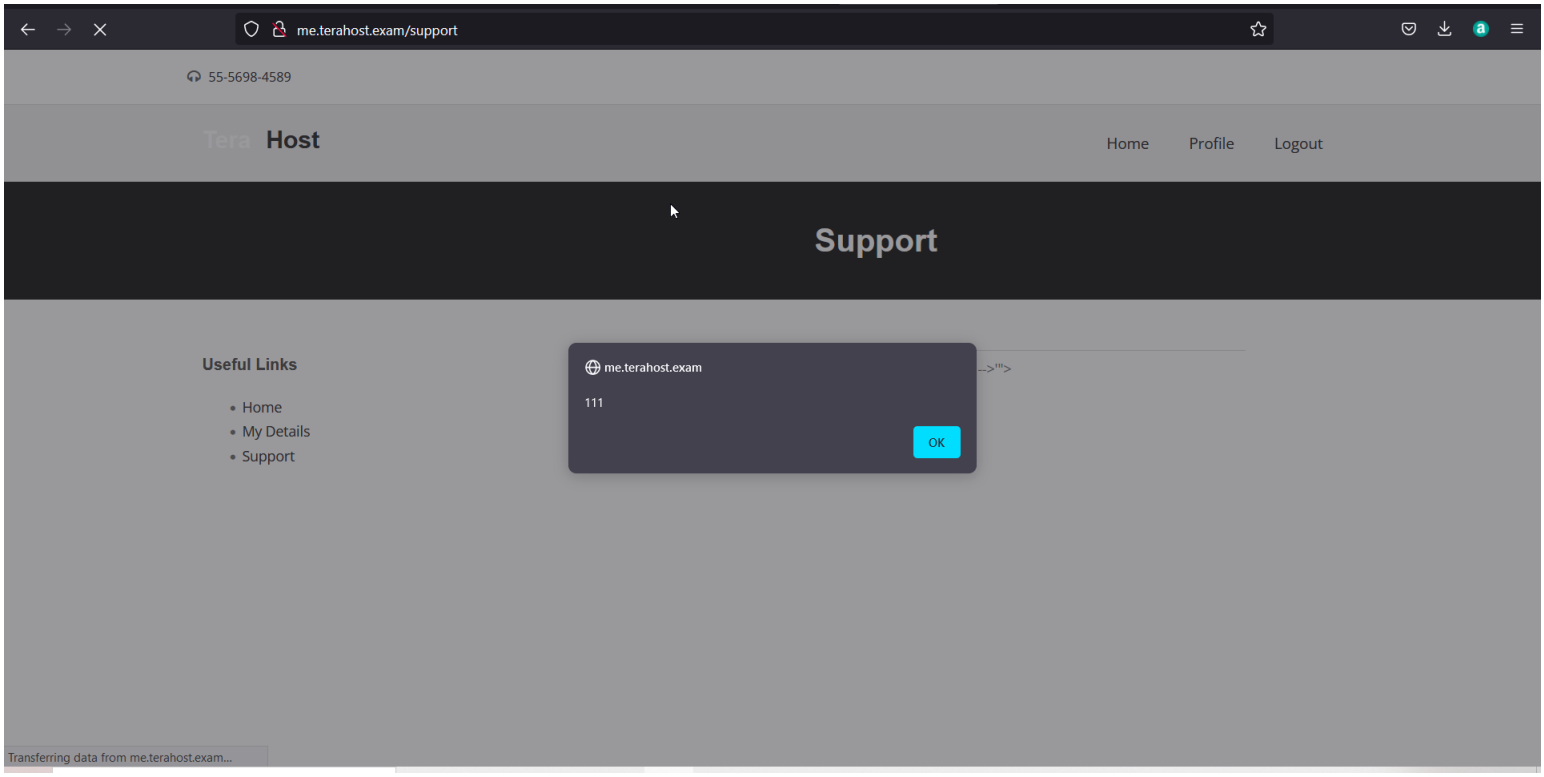
me.terahost.exam

111

OK

XSS in /support page

Open the /profile page and save the firstname with xss payload, now open me.terahost.exam/support page xss gets triggered.



XSS in articles parameter

Open blog.terahost.com and open any article , now replace the article parameter number with xss payload and load the page. Xss gets triggered .

Go Cancel

Target: http://blog.terahost.exam

Request

Raw Params Headers Hex

```
GET /?page=articles--%3E%3Cscript%3Ealert(111)%3C%2fscript%3E HTTP/1.1
Host: blog.terahost.exam
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:94.0) Gecko/20100101 Firefox/94.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: PHPSESSID=7p121hbsc4n8kmjo0p8e4fj173
Upgrade-Insecure-Requests: 1
```

Response

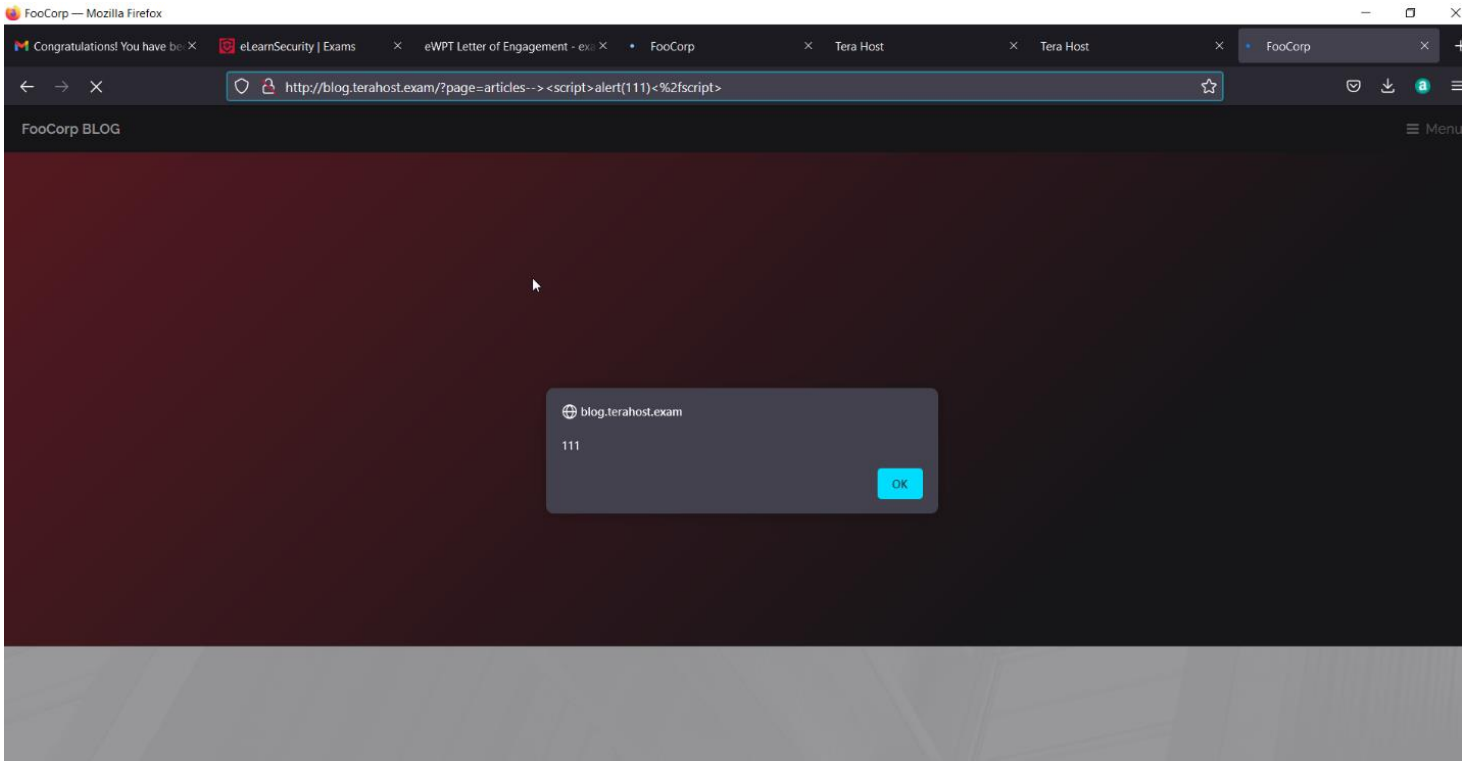
Raw Headers Hex HTML Render

```
<!-- Nav -->
<nav id="menu">
<ul class="links">
<li><a href="?page=index">Home</a></li>
<li><a href="?page=articles">Articles</a></li>
<!--<li><a href="?page=login">Log In</a></li>-->
</ul>
</nav>

<!-- Banner -->
<section id="banner">
<div class="inner">
<h1>FOOCorp BLOG</h1>
<p>Business & more... </p>
</div>
<video autoplay loop muted playsinline src="images/banner.mp4"></video>
</section>


<!-- Page articles--><script>alert(111)</script> does not exist!-->  <!-- Highlights -->
<section class="wrapper">
<div class="inner">
<header class="special">
<h2>Business articles</h2>
<p>Featured <a href="/?page=articles">ARTICLES</a> to read all news about
FooCorp.</p>
</header>
<div class="highlights">
<section>
<div class="content">
<header>
<a href="#" class="icon fa-vcard-o"><span
class="label">Icon</span></a>
<h3>12.12.2019 Remedy for sales decrease</h3>
</header>
<p><a href="/?page=articles&id=24">Read</a></p>
</div>

</section>
```



Recommendation:

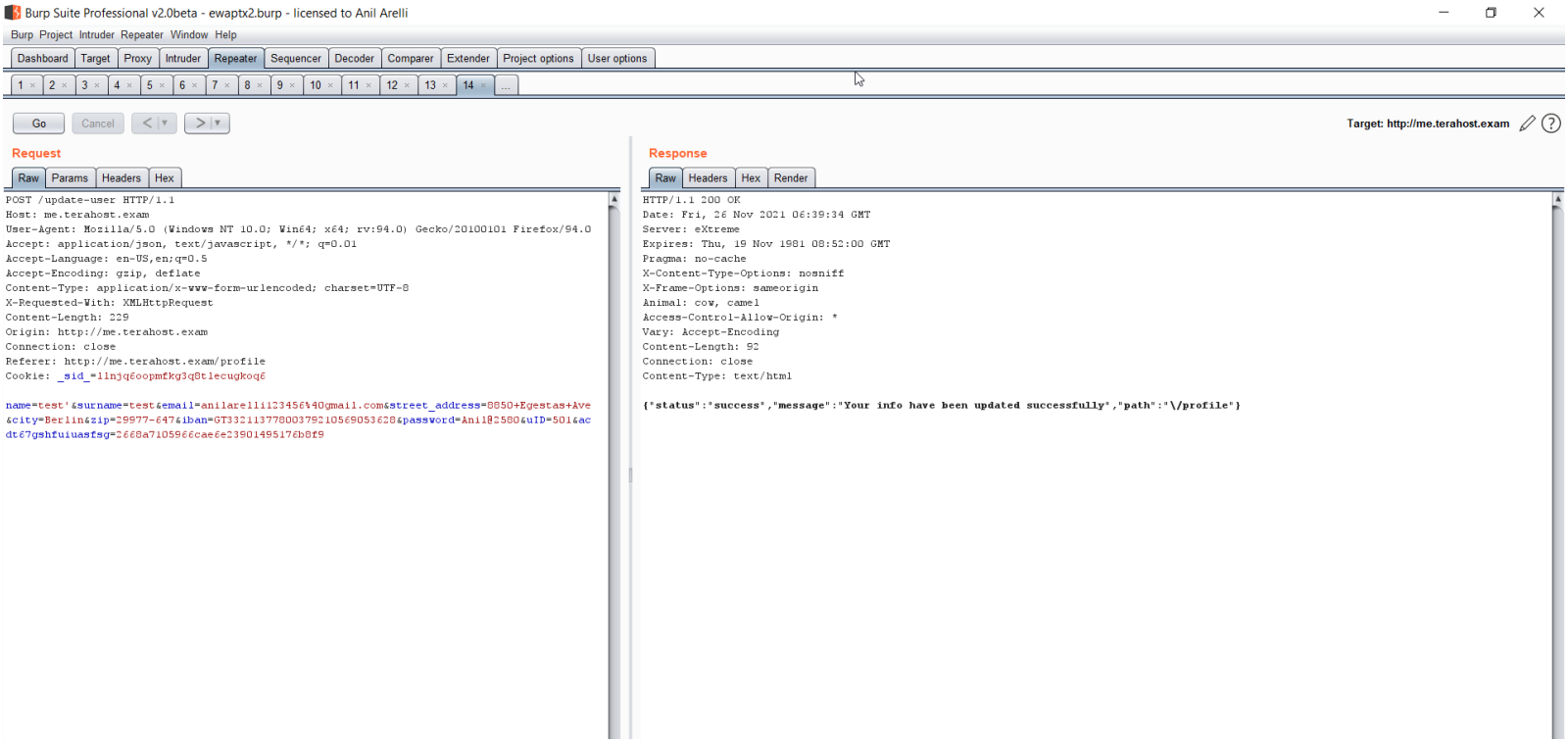
- Validation on user input fields
- Encode data on output
- Use appropriate response headers: Use Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.

4.2 privilege escalation / account takeover through IDOR.

Risk Rating: High

Analysis : Insecure direct object references (IDOR) are a type of access control vulnerability that arises when an application uses user-supplied input to access objects directly.

POC:- while updating user info from /profile page , It is possible for a user to change other users profile info like username , password, email, address by changing the uid parameter to victim uid.



Recommendation:

Access control vulnerabilities can generally be prevented by taking a defense-in-depth approach and applying the following principles:

- Never rely on obfuscation alone for access control.
- Unless a resource is intended to be publicly accessible, deny access by default.
- Wherever possible, use a single application-wide mechanism for enforcing access controls.
- At the code level, make it mandatory for developers to declare the access that is allowed for each resource, and deny access by default.
- Thoroughly audit and test access controls to ensure they are working as designed.

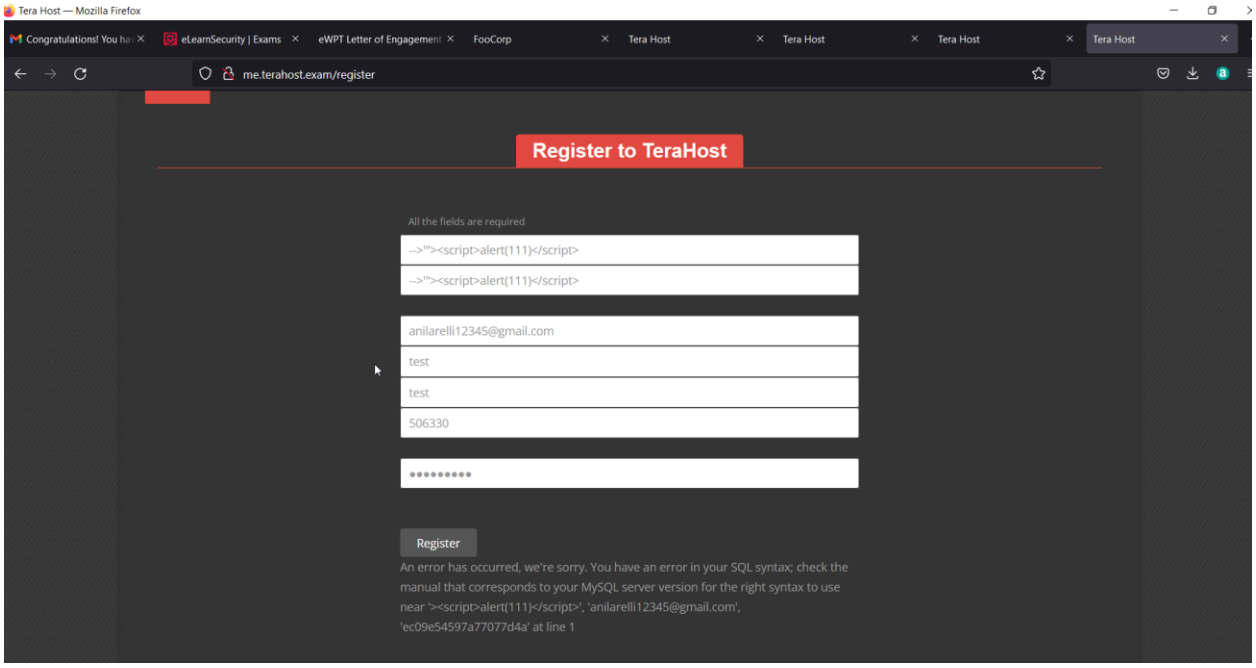
4.3 Sql injection

Risk Rating: High

Analysis : SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database.

POC:-

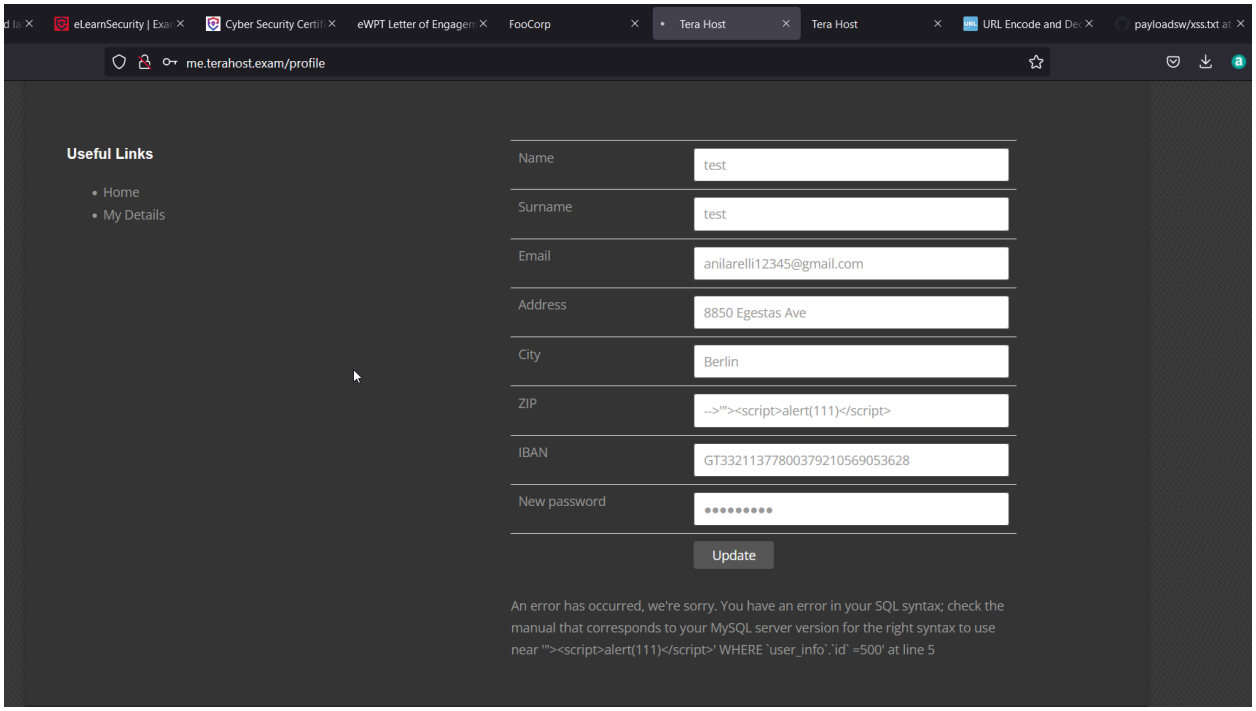
While registering enter single & double quotes in firstname field which will throw an sql error .



SQLI in /profile page

POC:-

In /profile page enter single & double quote in zip field with will trigger and sql error .



Remediation:

Most instances of SQL injection can be prevented by using parameterized queries (also known as prepared statements) instead of string concatenation within the query.

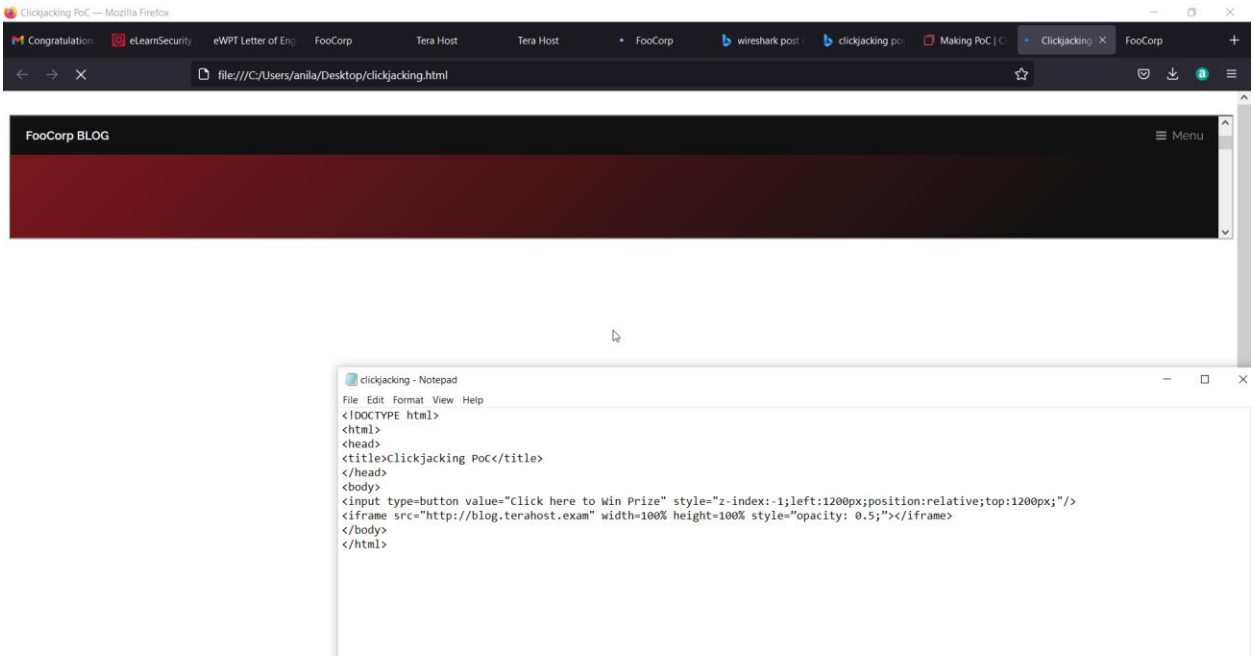
4.4 clickjacking

Risk Rating: Low

Analysis:

Clickjacking is an interface-based attack in which a user is tricked into clicking on actionable content on a hidden website by clicking on some other content in a decoy website. Consider the following example:

POC:



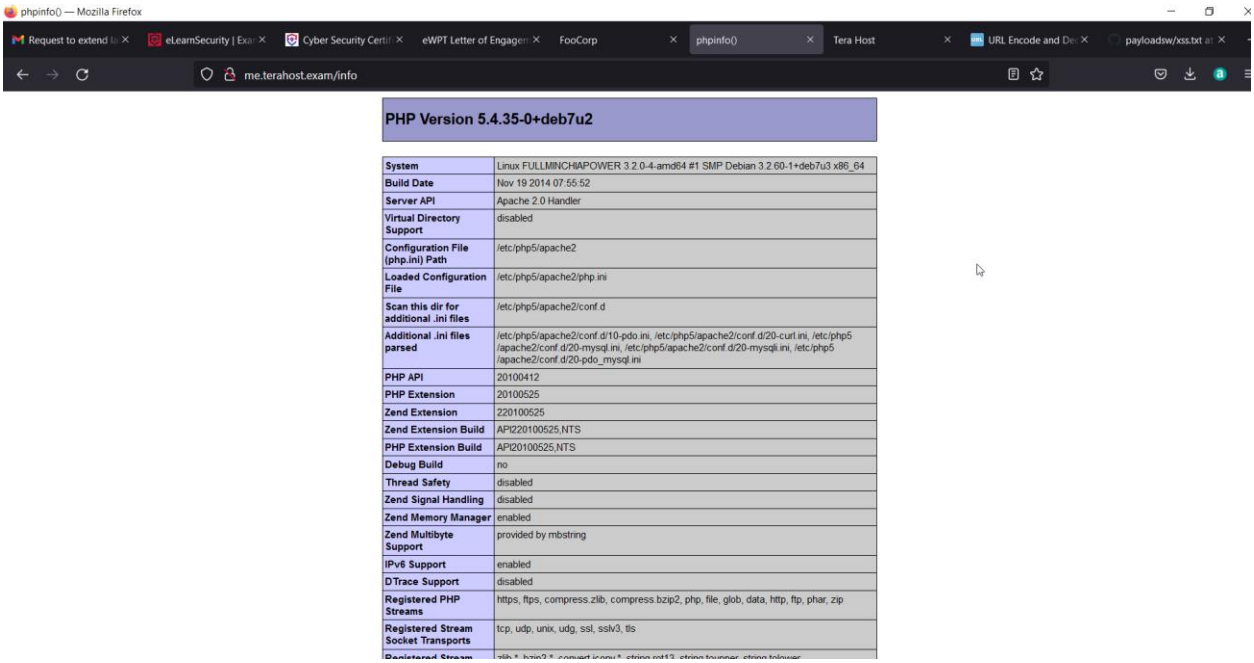
Remidiation :

Two mechanisms for server-side clickjacking protection are X-Frame-Options and Content Security Policy.

4.5 PHP info

Risk Rating: Low

POC :- php info page can be directly accessible by requesting me.terahost.exam/info



System	Linux FULLMINCHIAPOWER 3.2.0-4-amd64 #1 SMP Debian 3.2.60-1+deb7u3 x86_64
Build Date	Nov 19 2014 07:55:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls

Remediation:

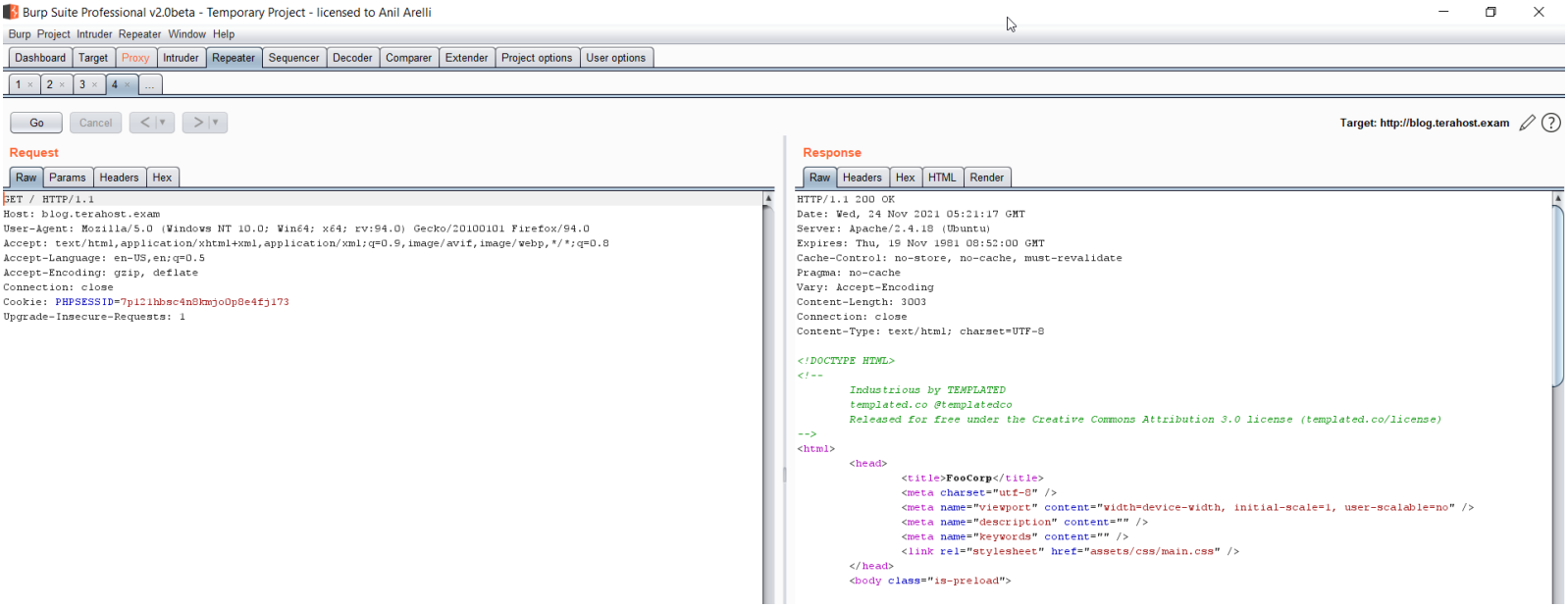
Remove this page as this might disclose sensitive information .

4.6 Missing security Header

Risk Rating: Info

Analysis:- It was found that there are no security header implemented

POC:-



Remidiation:

Implement below security Headers.

1. Cache-Control: no-cache,no-store

2. X-Content-Type-Options=nosniff

3. X-XSS-Protection: 1; mode=block

4. X-FRAME-OPTIONS: SAMEORIGIN

5. CSP / Content Security Policy

6. Strict-Transport-Security: max-age=86400; includeSubDomains

7. Implement set Secure Flag for Cookie.

8. Implement set HttpOnly flag for Cookie