Penetration testing Report of Company XYZ

Conducted By
MD Anwar Hossain

Executive Summary

This penetration testing is conducted completely in the “Black box” manner of the company XYZ web application.
The following sections provide overview of the vulnerabilities, recommendation and s
As per contract, social engineering and DDos attack out of scope. The engagement didn’t conduct these.
The rating of the testing. I highly recommend reviewing the section of Summary of business risks and High-Level Recommendations for better understanding of risks and discovered security issues.

Scope	Security Level	Grade
Web application	Poor	D

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.
E	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.

Vulnerability Summary

The test discovered a few vulnerabilities that may cause the broken confidentiality and integrity of the resources. Identified vulnerabilities easily exploitable by the attacker and application can be damaged.

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

Severity	Critical	High	Medium	Low	Informational
No of issues	3	2	3	1	2

Finging	Severity
FN-01 Database Username and password publicly available in github	Critical
FN-02 Information Disclosure via Http response header	Informational
FN-03 Open ports are showing via Port scanning	Informational
FN-04 Account profile can be changed by attacker from the edit profile section without interaction with user	High
FN-05 Host header Poisoning	Low
FN-06 Password Reset Poisoning	Critical
FN-07 IDOR attack to bypass normal user to super admin	Critical
FN-08 Vulnerable version of Jquery installed JQuery 1.2 < 3.5.0 Multiple XSS	Medium

FN-09 Nuked-Klan index.php Multiple Module Vulnerabilities	Medium
FN-10 TLS Version 1.0 Protocol Detection	Medium

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. 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.

Scope:

Assessment	Company Name	Asset Details
Penetration testing of Web application	XYZ	Website: *.test.xyz.com *.xyx.com IP Address: 192.168.0.1

I performed a discovery process to gather information about the assets and search for information disclosure vulnerabilities.I tested the authentication and authorization, session management, user input sanitization process. This penetration testing purpose is to mitigate the weakness of the application,so I recommended the mitigation strategies for improving the security posture.

Security tools Used

- Burp suite
- Nmap
- Gobuster
- DNSenum
- Nikto
- Nessus
- TestSSL
- Trufflehog
- Wfuzz

- Metasploit
- Shodan
- Crt.sh
- Sqlmap

Methodology

I followed the following methodology:

- OWASP Top 10 Application Security risks 2021
- OWASP testing guide WSTG v-4

The Open Web Application Security Project is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security. The Open Web Application Security Project provides free and open resources.

Findings Details

FN-01 Database Username and password publicly available in github.

Severity: Critical

Location:

- xyz.com

Issue Description:

Database administrators and developers use credentials for accessing the root user of the database. They keep the application users' details on the database. If it leaks, companies' reputation can be hampered.

Proof of vulnerability :

Use google dorking for searching github repositories. One repository has come in the result, which is the developer private repository. He put the database credentials in the github repository docker configuration file.

```
1  ##############################################################################
2  #                          DOCKER CONFIGURATION                              #
3  ##############################################################################
4
5  # ----------------------------
6  # APPLICATION CONFIGURATION
7  # ----------------------------
8  PHP_VERSION=7.2
9  APPLICATION_TIMEZONE=[redacted]
10 APPLICATION_RUNNING_PORT=8081
11 APPLICATION_RUNNING_HTTPS_PORT=8082
12 UID=1000
13 GID=1000
14 BUILD_MODE=dev
15 BUILD_TAG=latest
16 PHPMYADMIN_PORT=8085
17
18 # ----------------------------
19 # DATABASE CREDENTIALS
20 # ----------------------------
21 DATABASE_HOST=[redacted]
22 DATABASE_USERNAME=[redacted]
23 DATABASE_PASSWORD=[redacted]
24 DATABASE_ROOT_PASSWORD=[redacted]
25 DATABASE_PORT=3308
26 DATABASE_NAME=[redacted]
```

Impact: Hackers can login to the database and dump username and password .

Picture: Proof of the evidence of database credentials.

Recommendation:

1. Use gitleaks or trufflehog to check the github repository.
2. Be aware when pushing data on github.

FN-02 Information Disclosure via Http response header

Severity: Informational
Location:

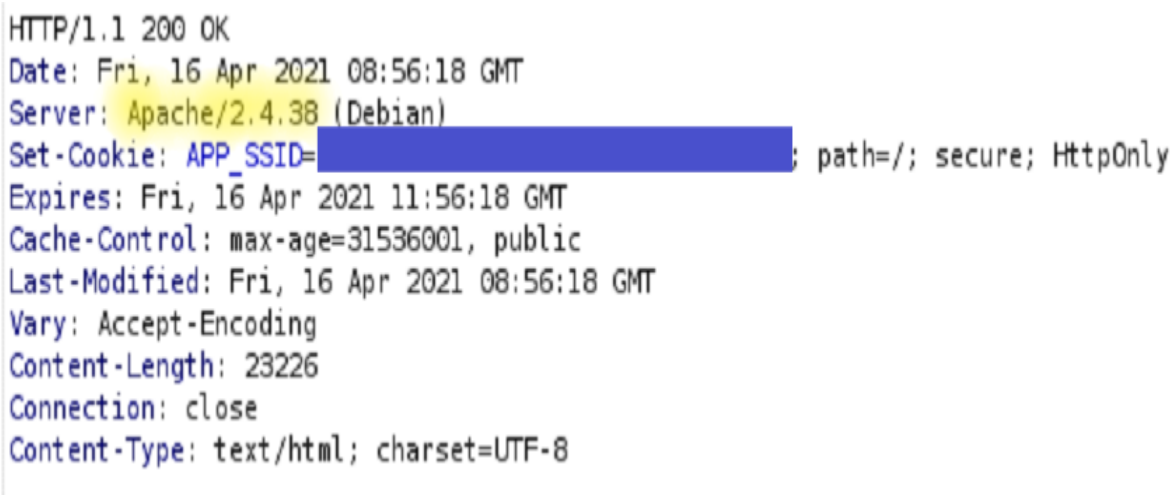
- xyz.com

Tools used: Burp suite

Issue description:
Revealing used application versions is risky , if any vulnerability is publicly discovered and published. It can lead the hacker to check the CVV or CVSS and can compromise the system.

Proof of vulnerability:

Browse the website and intercept the request by the burp suite. Check the response of the request. It revealed the version. The version is Apache 2.4.38 which is highlighted.



Recommendation:
The system administrator should show the version of the server.

FN-03 Open ports are showing via Port scanning

Severity: Informational
Location

- xyz.com

Issue description:
Open ports provide the attackers with the pathway of compromising the environment. It isn't immediately vulnerable. But, it becomes dangerous when the services are exploited and security vulnerabilities are exposed.

Proof of vulnerability:
Used the port scanning tools for scanning service and version. Found the ports and services.

Used command :
nmap -sC -sV -oA nmap ip

```
PORT     STATE  SERVICE       VERSION
22/tcp      open     ssh               OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; prot
| ssh-hostkey:
|   3072 67:c4:f1:28:c3:75:52:1e:e7:f2:19:3f:0f:b4:ed:dd (RSA)
|   256 ad:13:c5:58:55:47:29:15:9c:19:b1:62:e0:99:2b:b6 (ECDSA)
|_  256 d5:4b:83:a5:ce:22:68:64:0a:c5:b7:a8:2d:0f:23:c4 (ED25519)
80/tcp   open   http          Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: [redacted]
443/tcp  open   ssl/http      Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: 400 Bad Request
| ssl-cert: Subject: [redacted]
| Subject [redacted]
| Not valid before: 2021-03-27T14:22:20
|_Not valid after:  2021-06-25T14:22:20
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
3389/tcp closed ms-wbt-server
5000/tcp closed upnp
8001/tcp closed vcom-tunnel
8002/tcp open   http          Apache httpd 2.4.38 ((Debian))
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: phpMyAdmin
8007/tcp closed ajp12
8008/tcp closed http
8009/tcp closed ajp13
8010/tcp closed xmpp
9200/tcp closed wap-wsp
```

Port: 22, 80,443, 8002 are opened.
It's the production site.

FN-04 Account profile can be changed by attacker from the edit profile section without interaction with user

Severity: High
Location:

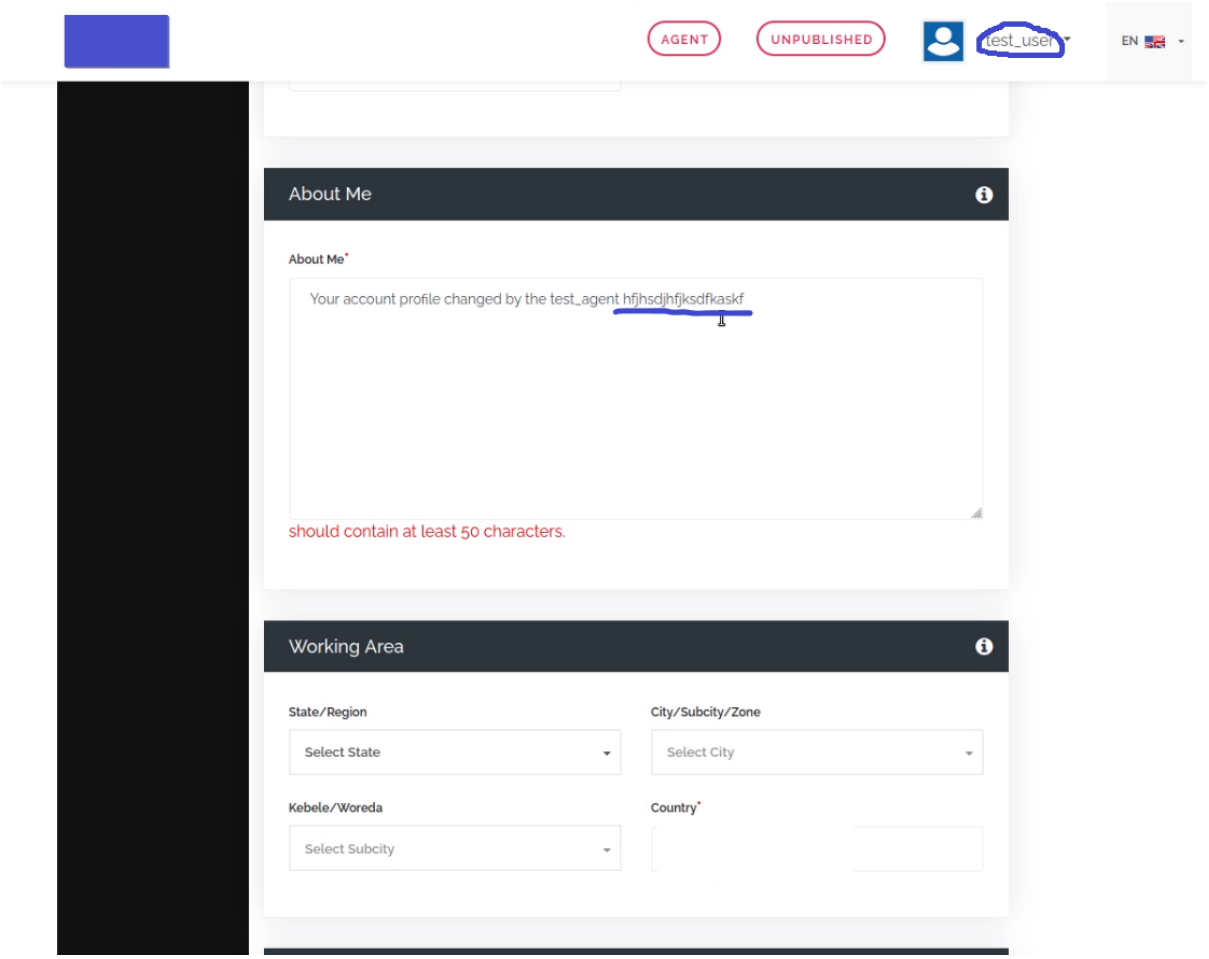
- xyz.com

Issue description:

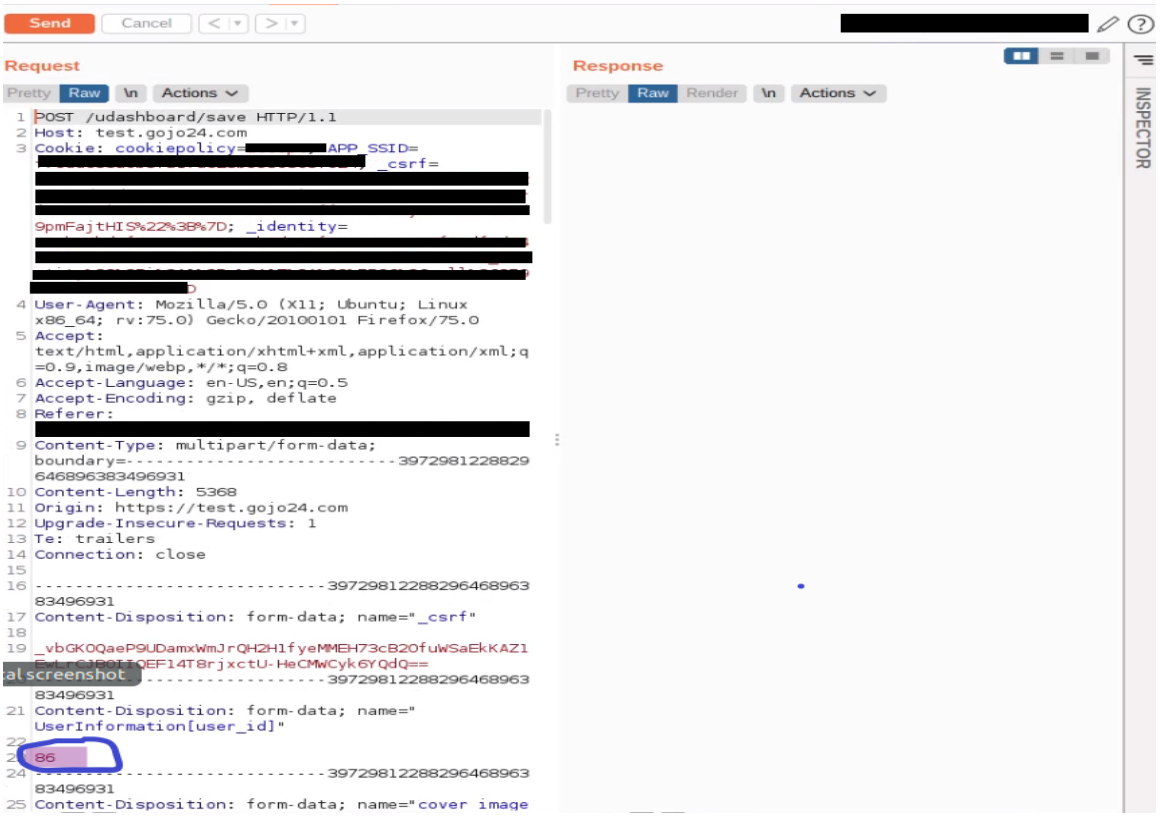
Account takeover (ATO) or account hijacking is an attack which allows an attacker to gain the access of the target user.

Proof of vulnerability:

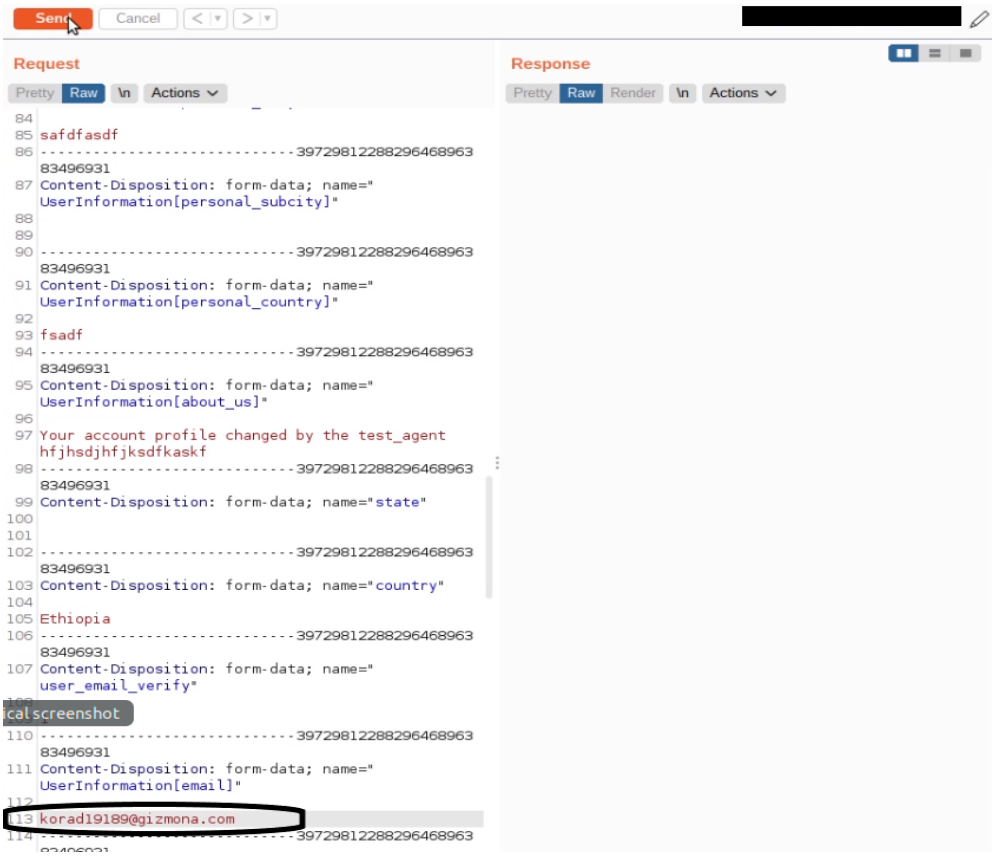
Opened two accounts for checking the vulnerability. One is the victim account and another one is the attacker.



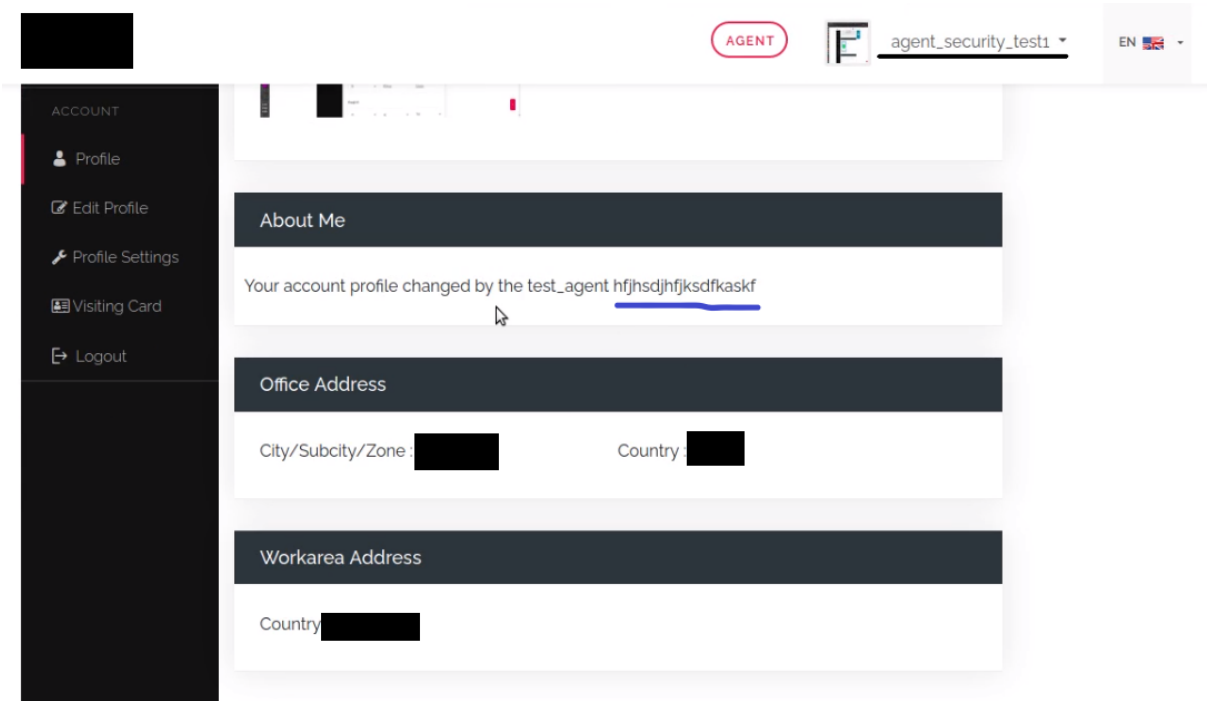
Intercept the request of the attacker.



User id shows here. Now change the user_id with the victim user_id. Change the mail address with one-time mail. The victim account about us section would change.



Victim account screenshot attached here.



Recommendation:
Validate user input before validation.

FN-05 Host header Poisoning

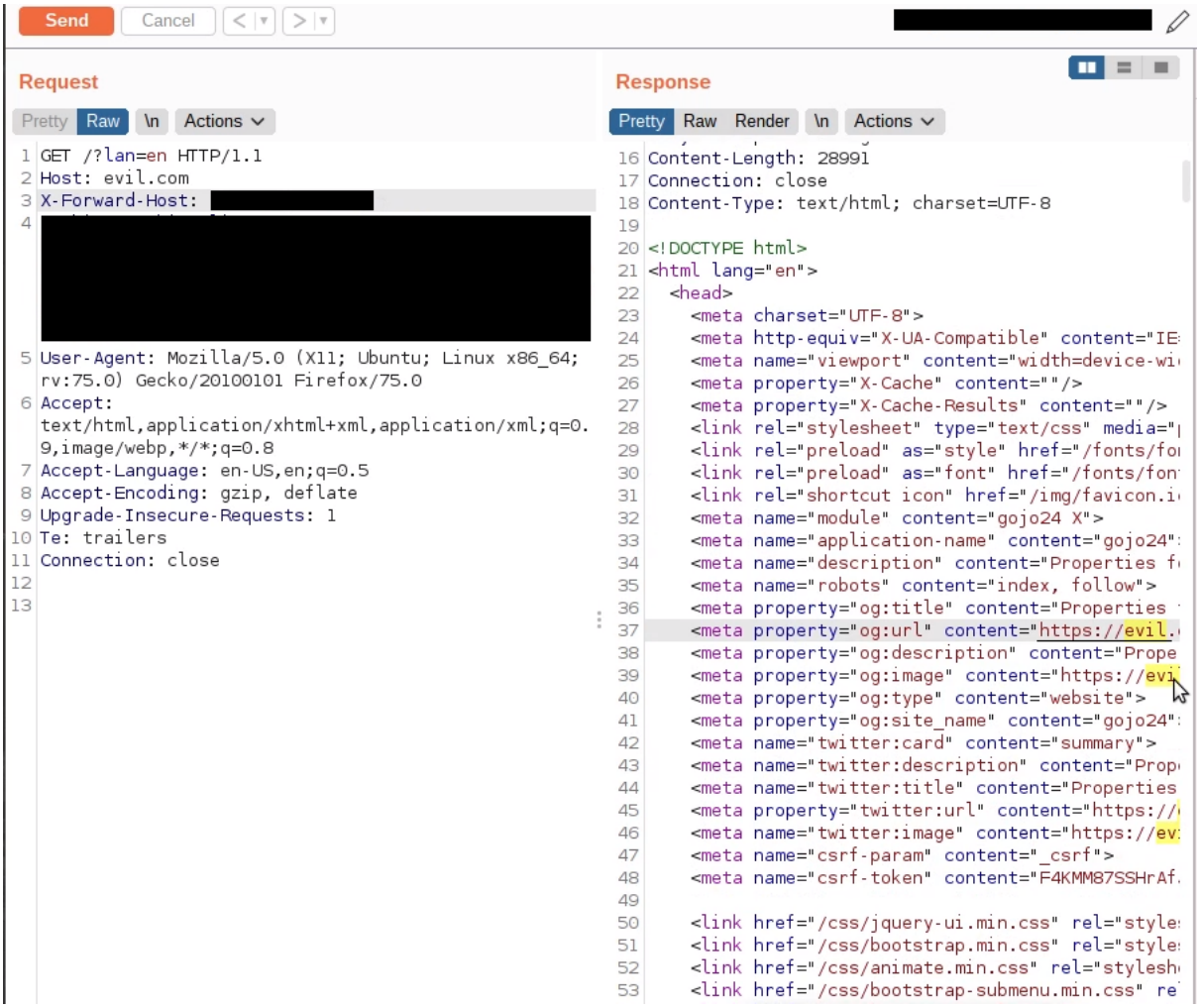
Severity: Low

Location:

- xyz.com

Issue description:
The application appears to trust the user-supplied host header. By supplying a malicious host header with a password reset request, it may be possible to generate a poisoned password reset link. Consider testing the host header for classic server-side injection vulnerabilities. Depending on the configuration of the server and any intervening caching devices, it may also be possible to use this for cache poisoning attacks.

Proof of vulnerability:
Change the host with evil.com. It reflects on the response. When a user clicks on the link that redirects to evil.com.



Recommendation:

Don't trust the host header. In case of necessity of using the host header as a mechanism for identifying the location of the web server, it's highly advised to make use of a whitelist of allowed hostnames.

FN-06 Password Reset Poisoning

Severity: Critical

Location:

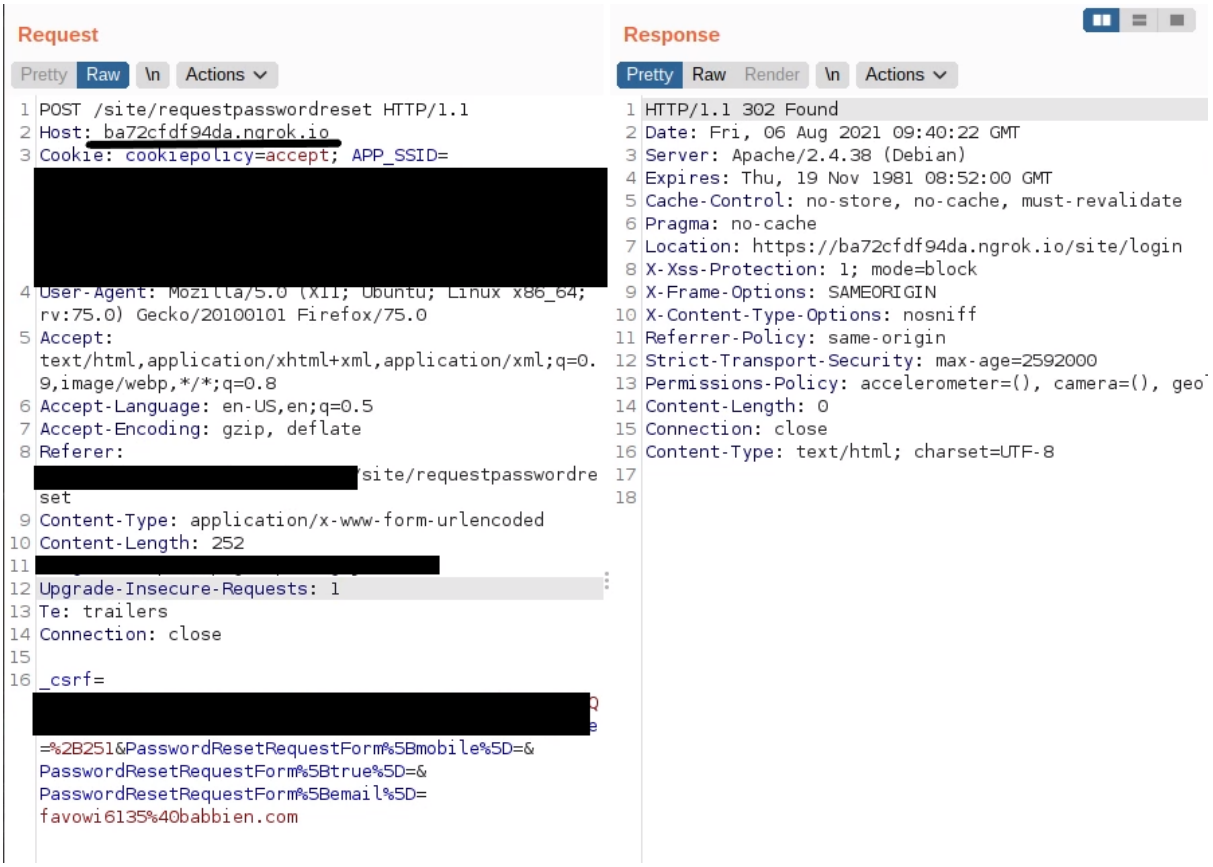
- xyz.com

Issue description:

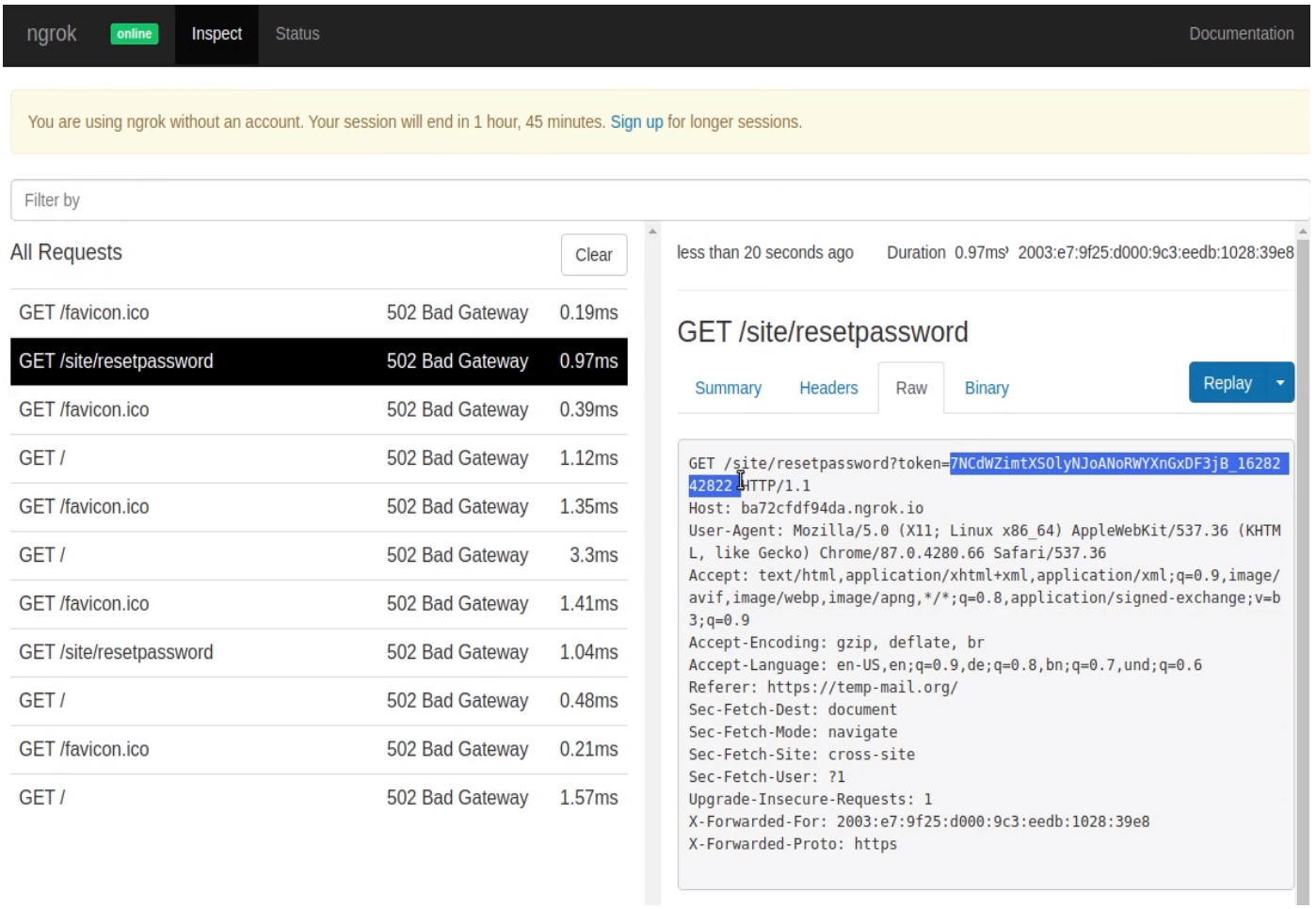
Password reset poisoning is a technique whereby an attacker manipulates a vulnerable website into generating a password reset link pointing to a domain under their control. This behavior can be leveraged to steal the secret tokens required to reset arbitrary users' passwords and, ultimately, compromise their accounts.

Proof of vulnerability:

Put the email address in the forgot password option. Send it and intercept the request through the burp suite. Change the host to an attacker hosted address which is created by ngrok.



After changing the request forwarded it. The user will get the link and if the user clicks the link then the attacker will get the password reset token. Attackers can change the password.



FN-07 IDOR attack to bypass normal user to super admin

Severity: Critical

Location:

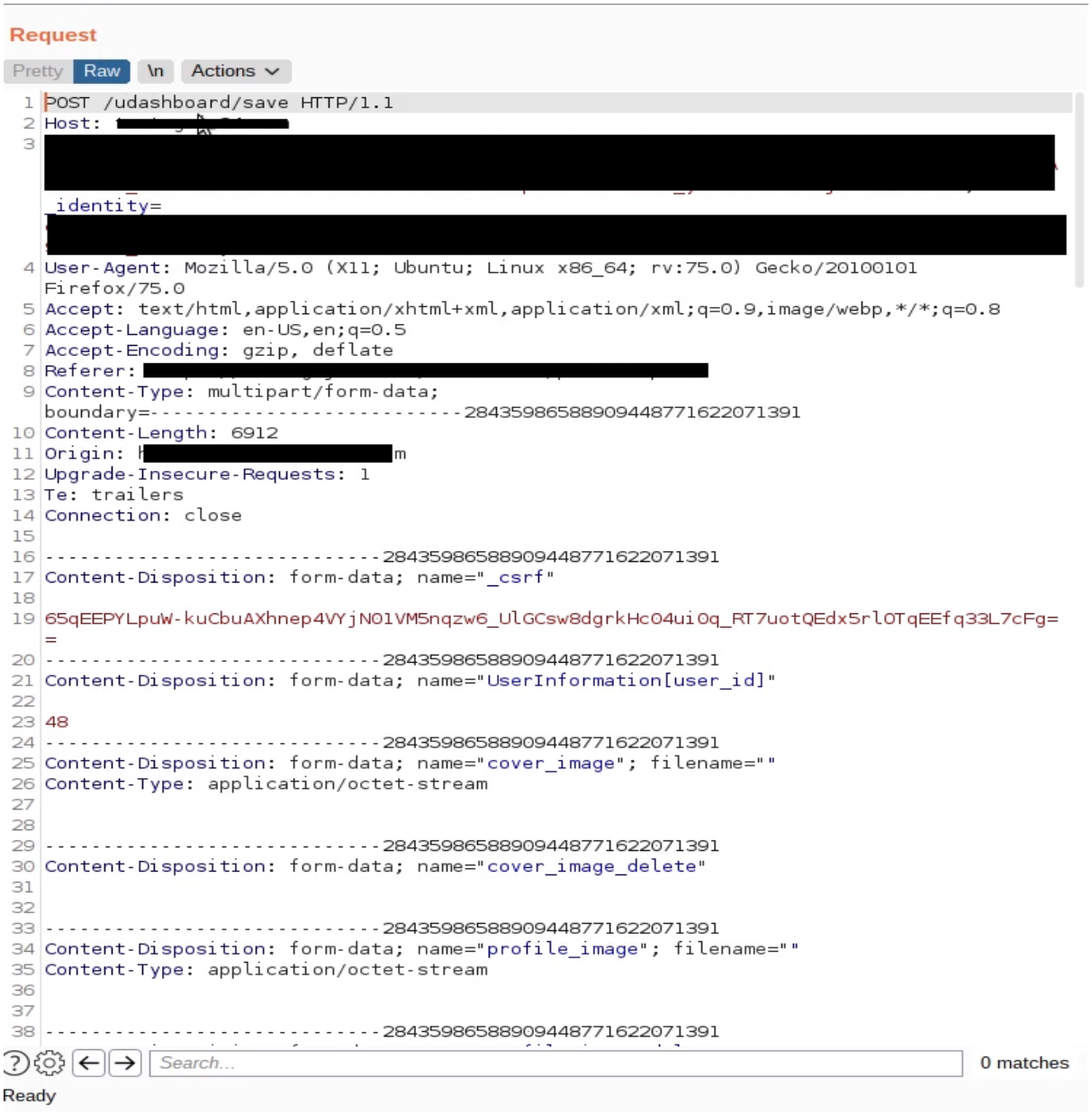
- xyz.com

Issue description:

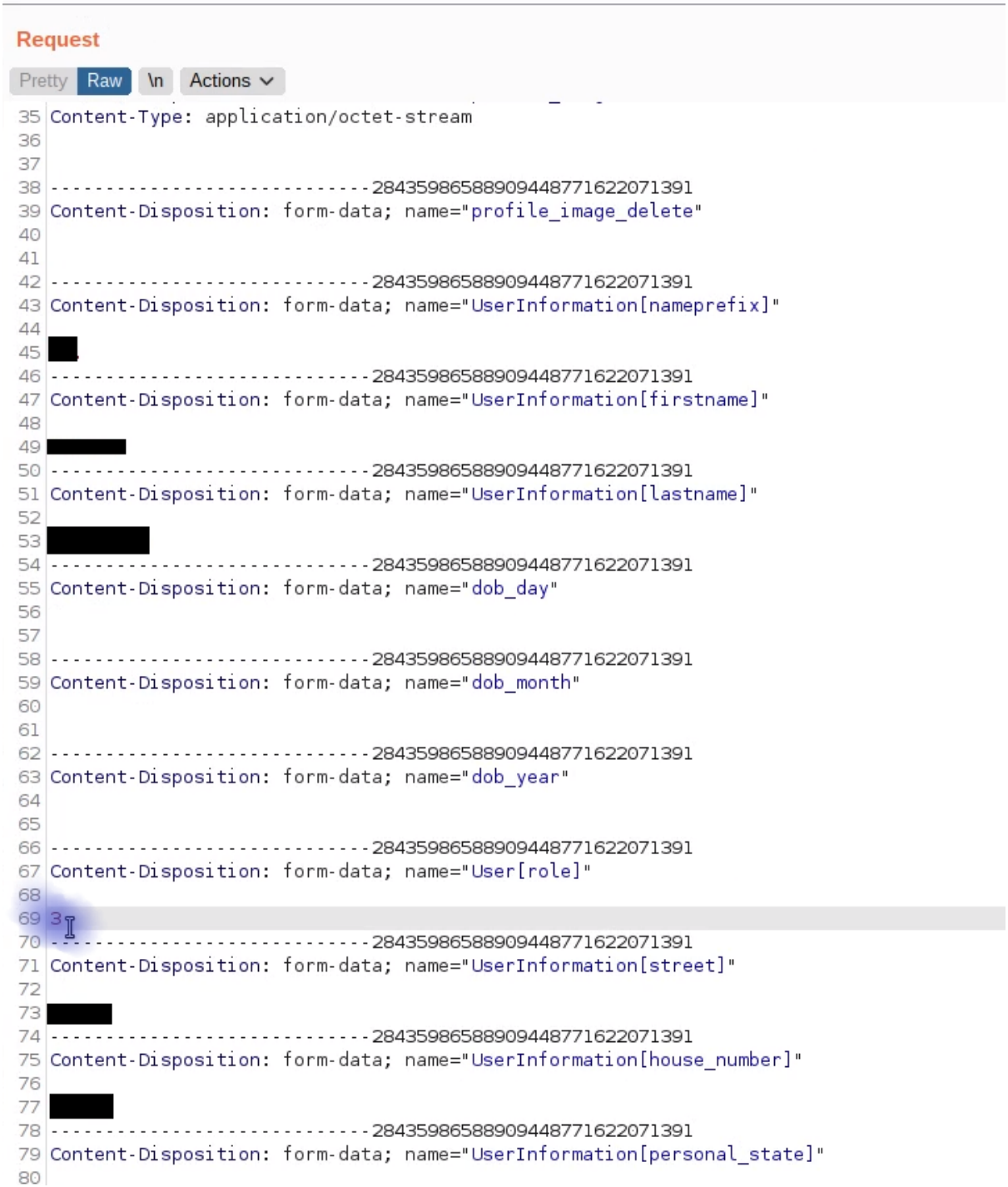
Insecure Direct Object Reference (called **IDOR** from here) occurs when an application exposes a reference to an internal implementation object. Using this way, it reveals the real identifier and format/pattern used of the element in the storage backend side. The most common example of it (although is not limited to this one) is a record identifier in a storage system (database, filesystem and so on).

Proof of vulnerability:

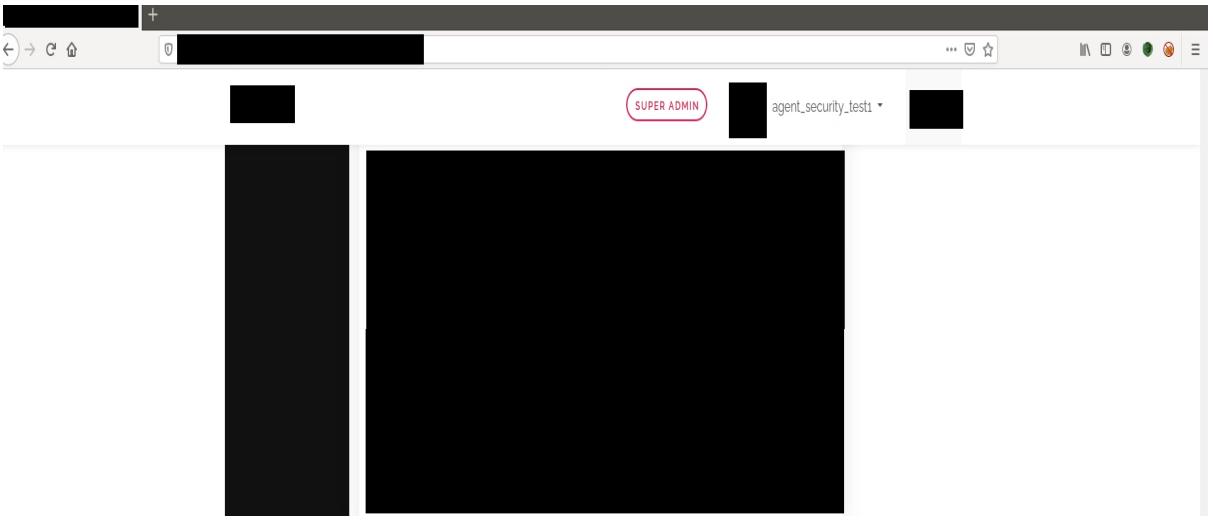
Create a normal user account. Change the edit profile, capture the request through burp suite. Observe the request and see there is a user_role variable where id is 3. Change the role_id 3 to 1 and put an email.



The role id has been changed in the below picture.



Then forward the request , then the user right elevated to the super user.



Recommendation:

1. Avoid predictable references.
2. Always validate user requests.

Reference
https://crashtest-security.com/insecure-direct-object-reference-idor/

FN-08 Vulnerable version of Jquery installed JQuery 1.2 < 3.5.0 Multiple XSS

Severity: Medium

Location:

- xyz.com

Issue description:
According to the self-reported version in the script, the version of JQuery hosted on the remote web server is greater than or equal to 1.2 and prior to 3.5.0. It is, therefore, affected by multiple cross site scripting vulnerabilities.

Note, the vulnerabilities referenced in this plugin have no security impact on PAN-OS, and/or the scenarios required for successful exploitation do not exist on devices running a PAN-OS release.

Proof of vulnerability:

```
URL               :                                                /assets/4bca7b27/jquery.js
Installed version : 3.4.1
Fixed version     : 3.5.0
```

Recommendation:
Upgrade to JQuery version 3.5.0 or later.

FN-09 Nuked-Klan index.php Multiple Module Vulnerabilities

Severity: Medium
Location:

- xyz.com

Issue description:

The instance of Nuked-klan running on the remote web server is affected by multiple vulnerabilities due to a failure to sanitize user-supplied input to several parameters before using them in the 'Team', 'News', and 'Liens' modules to display dynamic HTML. An unauthenticated, remote attacker can exploit these issues to execute arbitrary script code in a user's browser session.
Additionally, an information disclosure vulnerability exists that allows a remote attacker to disclose the physical path of the directory in which the application is installed; however, Nessus did not test for this.

Proof of vulnerability:

Recommendation:

- Patch with the stable version.

FN-10 TLS Version 1.0 Protocol Detection

Severity: Medium
Location:

- xyz.com

Issue description:
The remote service accepts connections encrypted using TLS 1.0. TLS 1.0 has a number of cryptographic design flaws. Modern implementations of TLS 1.0 mitigate these problems, but newer versions of TLS like 1.2 and 1.3 are designed against these flaws and should be used whenever possible.

As of March 31, 2020, Endpoints that aren't enabled for TLS 1.2 and higher will no longer function properly with major web browsers and major vendors.

PCI DSS v3.2 requires that TLS 1.0 be disabled entirely by June 30, 2018, except for POS POI terminals (and the SSL/TLS termination points to which they connect) that can be verified as not being susceptible to any known exploits.

Proof of vulnerability:

```
TLSv1 is enabled and the server supports at least one cipher.
```

Recommendation:
Enable support for TLS 1.2 and 1.3, and disable support for TLS 1.0.