

MAGMA FINCORP

WEB APPLICATION SECURITY ASSESSMENT REPORT

Milaap New Hire Joining Website

4 MAR 2021

INDEX

EXECUTIVE SUMMARY	3
ABOUT THE APPLICATION	5
VULNERABILITY LIST	6
VULNERABILITIES ILLUSTRATION	7

EXECUTIVE SUMMARY

SynRadar was assigned to conduct the security testing of 'Milaap' application by MAGMA FINCORP on 3rd March 2021. This security report illustrates the inferences gathered from the same. The security testing was carried out as per the standard Web application security testing methodology of SynRadar.

Scope of Testing: To perform security testing of the 'Milaap' web application.

URL - <https://milaap.magma.co.in/admin/login.php>

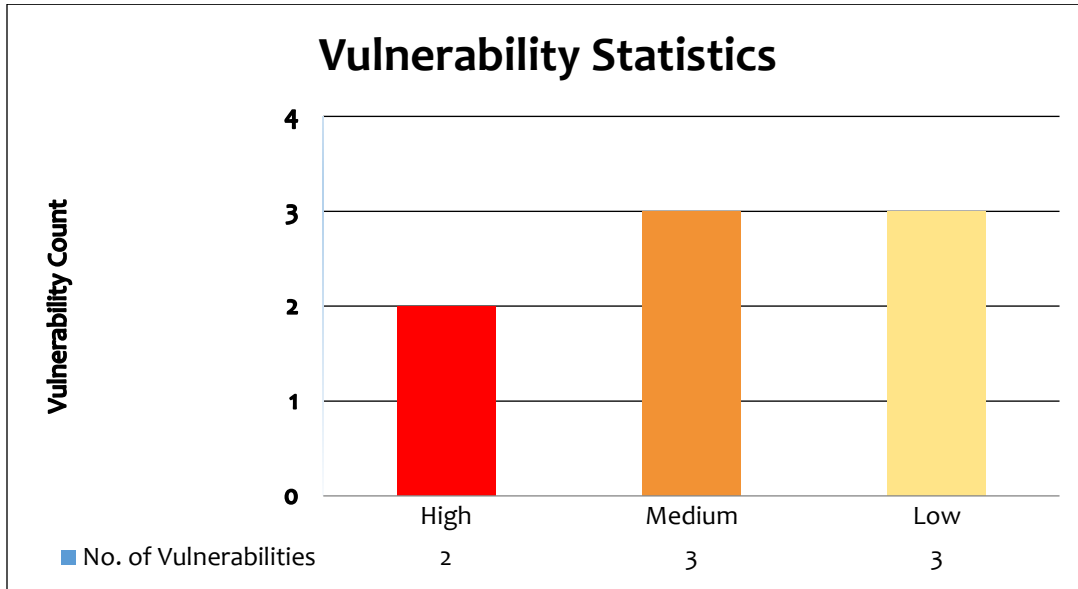
Inclusions: Whole application.

Authorization vulnerabilities will not be taken under consideration, As User Credentials were not provided for the testing. The testing was done on Admin Credentials.

Current Vulnerability Statistics:

The figure below represents the distribution of vulnerabilities in terms of High, Medium & Low severity issues found to be open in the application, as per this report.

Total number of vulnerabilities currently open are **8**.



ABOUT THE APPLICATION

- Milaap – New Hire joining kit.
- Candidates list, Upload IFSC data, Add User

VULNERABILITY LIST

Sr.no	VULNERABILITIES	RISK RATING
1	The application is vulnerable to Cross-site request forgery attack	High
2	The application does not implement authentication checks for downloads	High
3	Directory listing is enabled on the server	Medium
9	Application is vulnerable to Back refresh attack	Medium
6	The application has auto-complete feature enabled	Medium
4	The application does not implement required session cookie attribute	Low
7	Application using vulnerable libraries	Low
8	Application is vulnerable to Clickjacking	Low

VULNERABILITIES ILLUSTRATION

1. The application is vulnerable to Cross-site request forgery attack

HIGH RISK

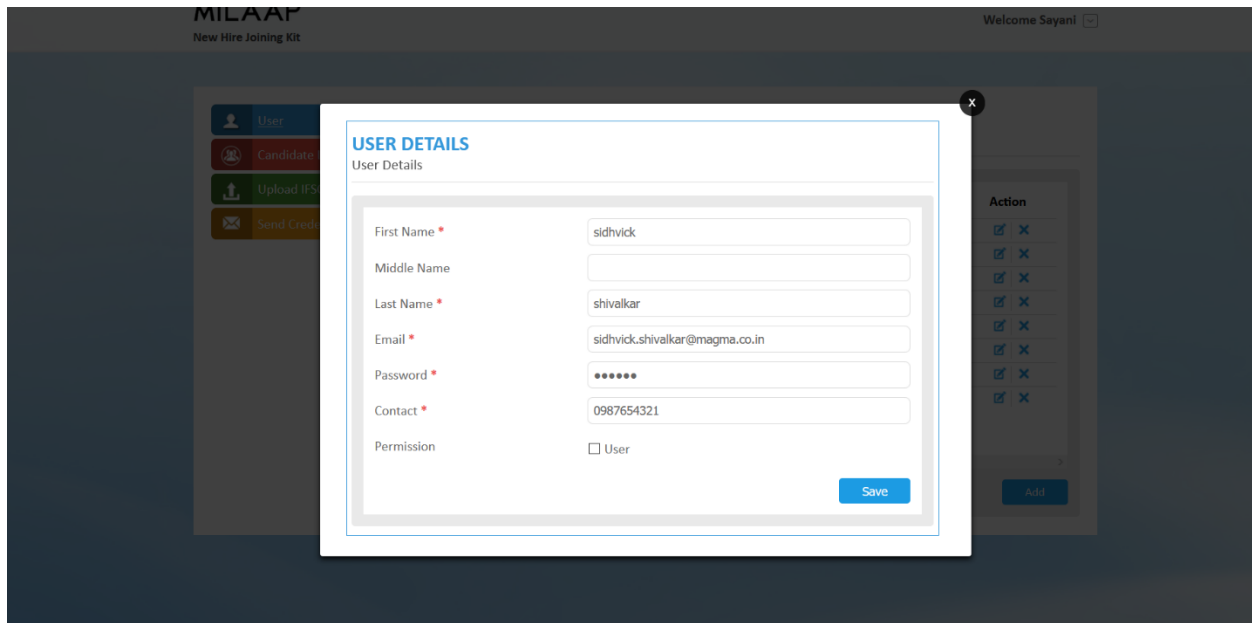
Description:

Cross-Site Request Forgery attack is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. An attacker can trick the victim into submitting a malicious request. It inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf. Therefore, if the user is currently authenticated to the site, the site will have no way to distinguish between the forged request sent by the victim and a legitimate request sent by the victim.

Vulnerable Location:

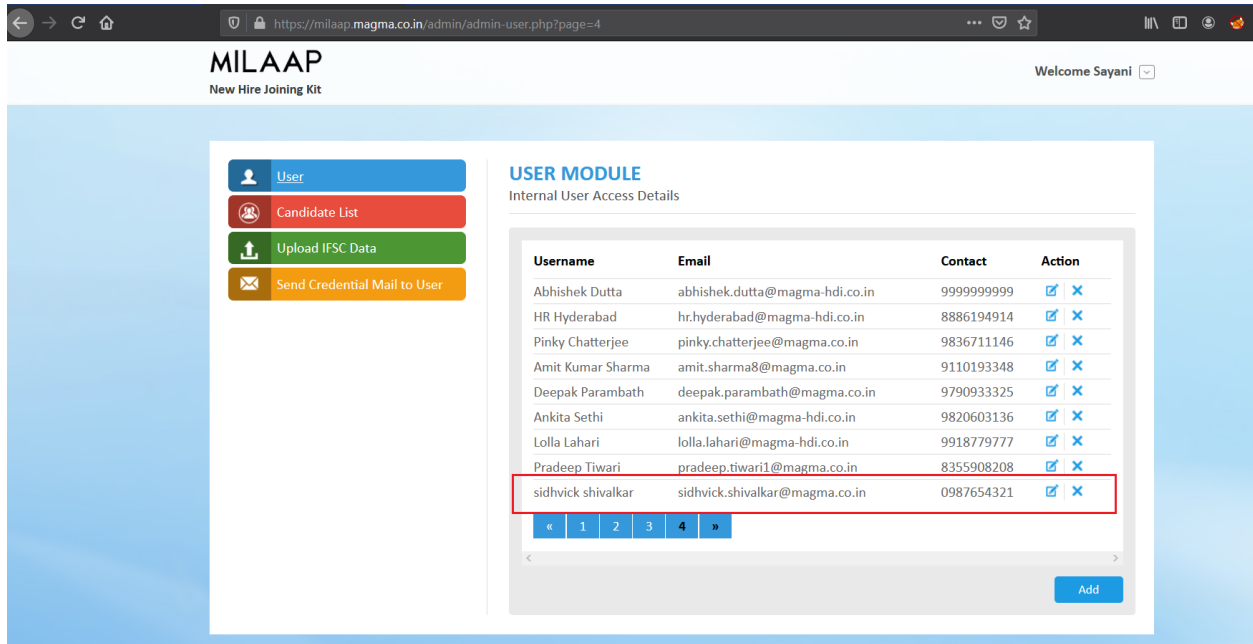
Throughout Application

Proof of concept: **Step 1:** A new user was added.

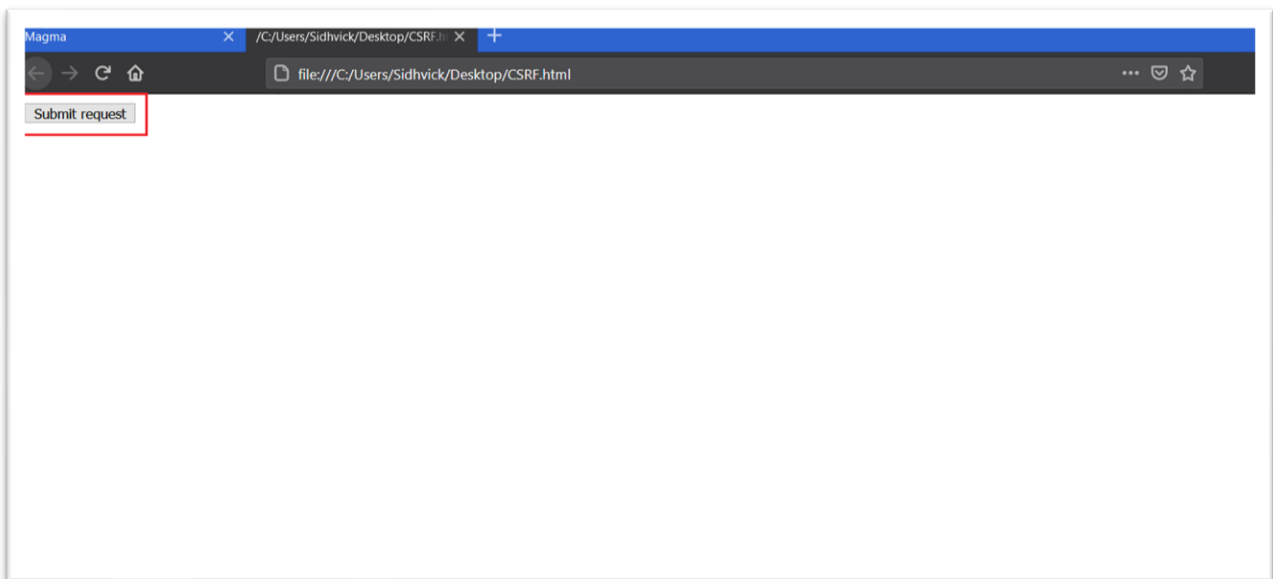


WEB APPLICATION SECURITY ASSESSMENT

Milaap New Hire Joining

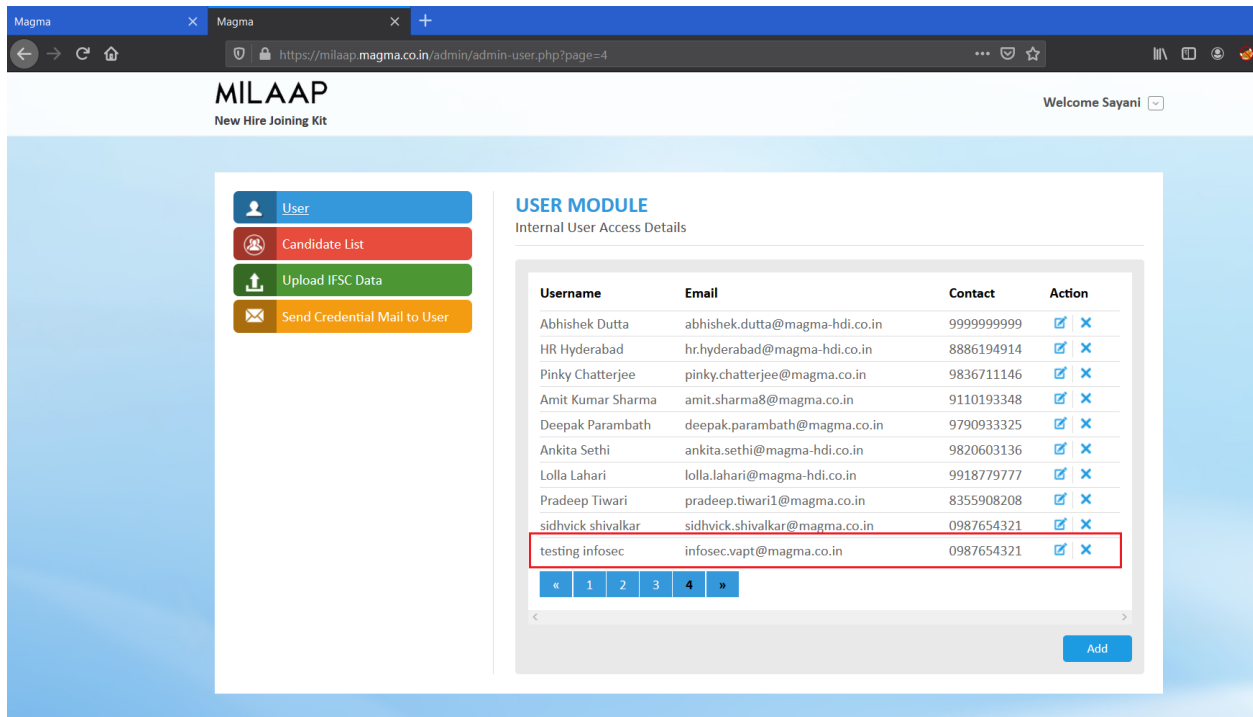


Step2: By creating a CSRF-PoC of the request the attacker can add users to the User Module.



WEB APPLICATION SECURITY ASSESSMENT

Milaap New Hire Joining



- The attacker can ADD, EDIT, DELETE users through cross site request forgery (CSRF)

Solution:

- Check standard headers to verify the request is from same origin
- Generate a security token and send the token within a hidden value in each form.
- Once the user submits the form, validate the submitted request based on the token value. Do not process the request without or with invalid token values.
- Invalidate the token and generate a new token for every request.
- For ASP.NET, use Anti-CSRF tokens.

Reference:

<https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/>

2. The application does not implement authentication checks for downloads

HIGH RISK

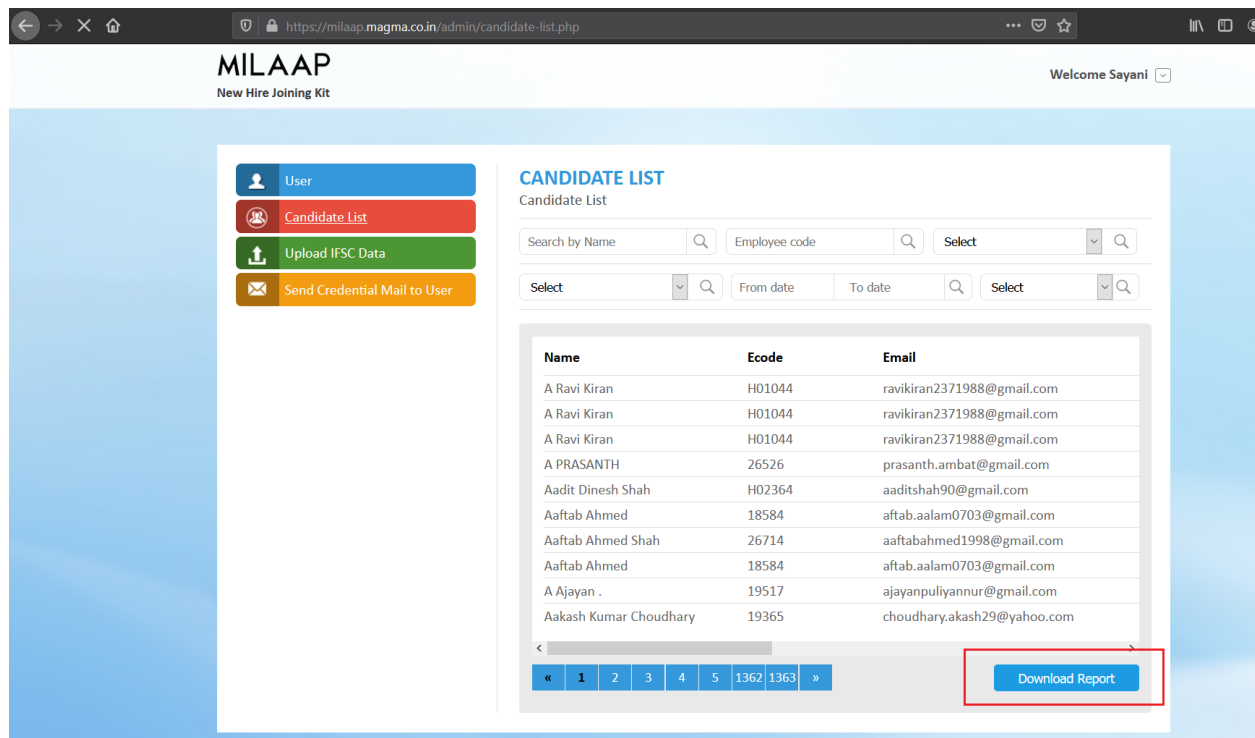
Description

It was observed that file which is having sensitive data like Email ID, Employee Code, Mobile number can be downloaded without any authentication, and the file can be downloaded through the vulnerable 'Download Report' Function.

Vulnerable URL:

<https://milaap.magma.co.in/admin/export.php?page=report>

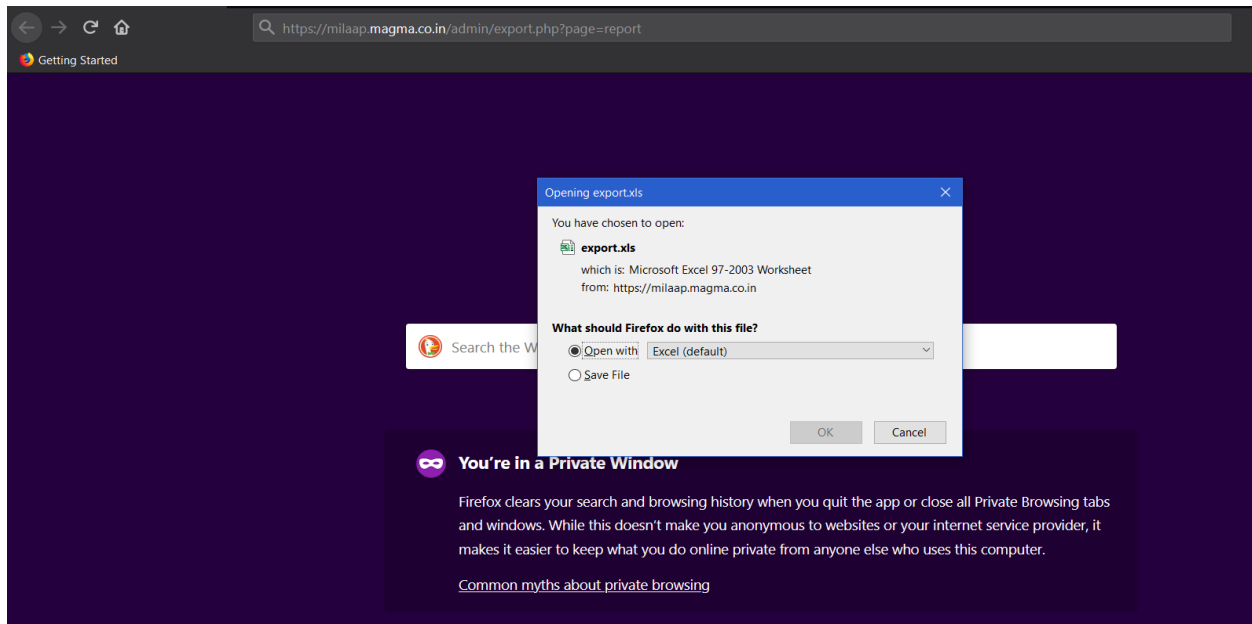
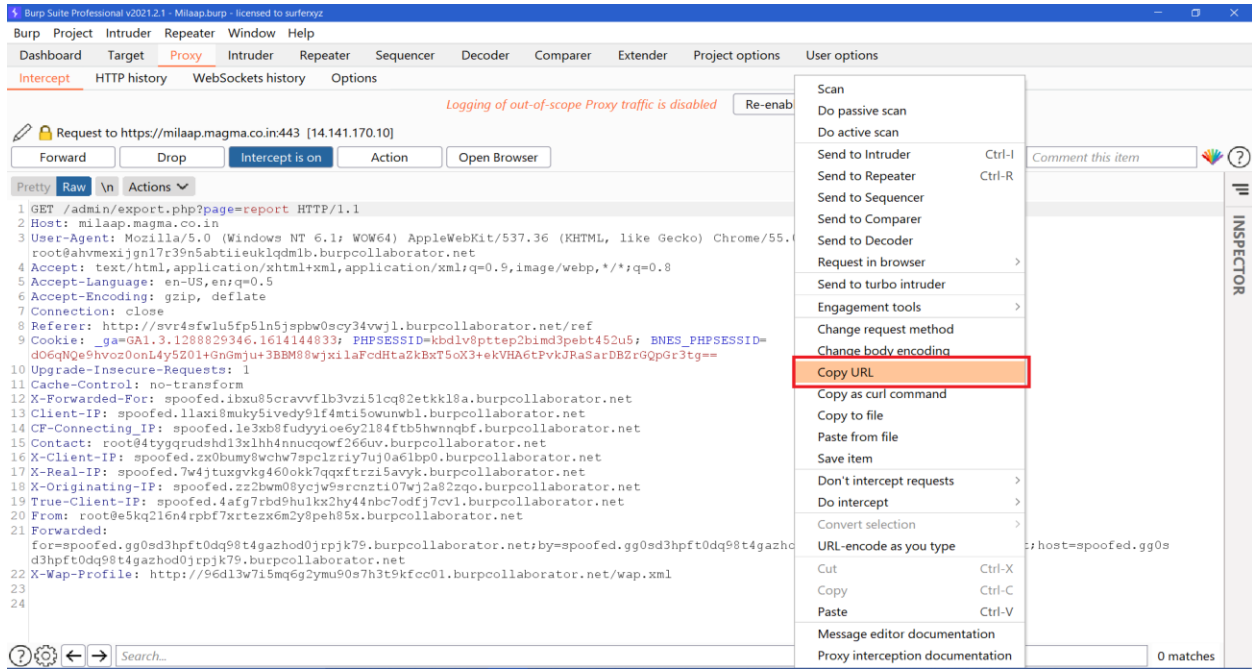
Proof of Concept: Step 1 – Capture the 'Download Report' request in Burp Suite.



Step 2 - Copy URL of the captured request, Paste it in any other browser or in incognito mode, the file will get downloaded without authentication.

WEB APPLICATION SECURITY ASSESSMENT

Milaap New Hire Joining



Solution: Implement authentication check before rendering internal files like Candidate List report.

3. Directory listing is enabled on the server

MEDIUM RISK

Description:

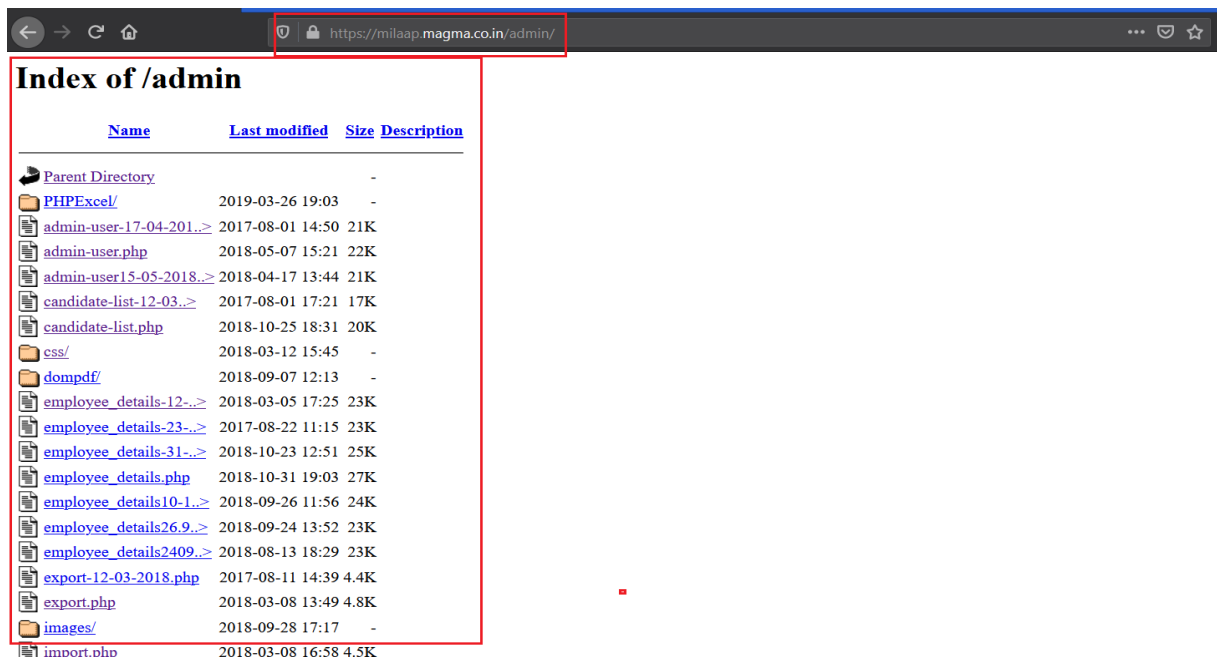
A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers. Directory listing provides an attacker with the complete index of all the resources located inside of the directory. The specific risks and consequences vary depending on which files are listed and accessible.

Vulnerable Location:

Throughout the application

- <https://Milaap.magma.co.in/admin/>

Proof of Concept:



Solution:

Disable Directory Listing on the server

4. Application is vulnerable to back refresh attack.

MEDIUM RISK

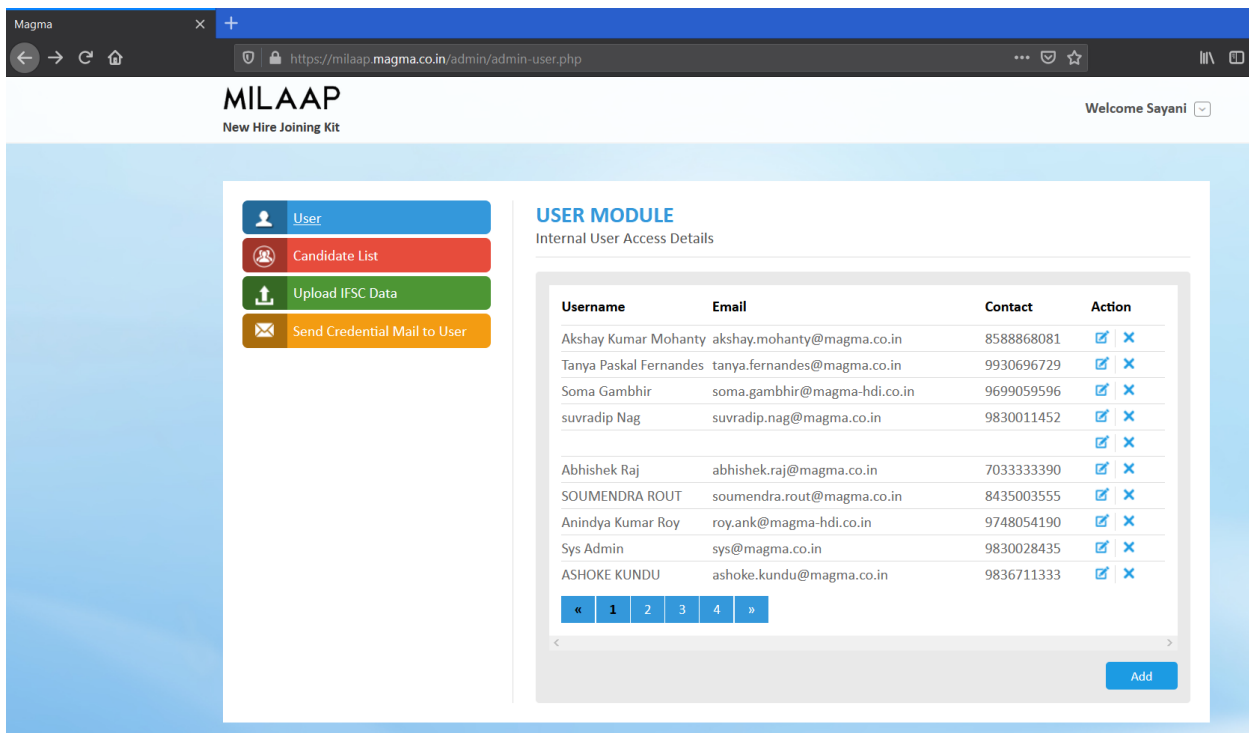
Description:

Back - refresh attack is an attack which enables an adversary to obtain application credentials by going back to previous page and re-submitting the expired-document.

Here, the application does not implement URL redirection after processing the login request. For instance, a user logs in to the application, performs some action and logs out of the application. If an attacker has access to the same machine, he can click 'Back' button until he reaches the page shown after a successful login. The attacker can then click 'Refresh' button, and the browser will automatically resubmit the request with all the information which can be stolen by the attacker using application layer proxy. Also, an attacker can get re-authenticated to the server.

POC:

Step 1: Login through the portal, after logging in press back button on your browser.

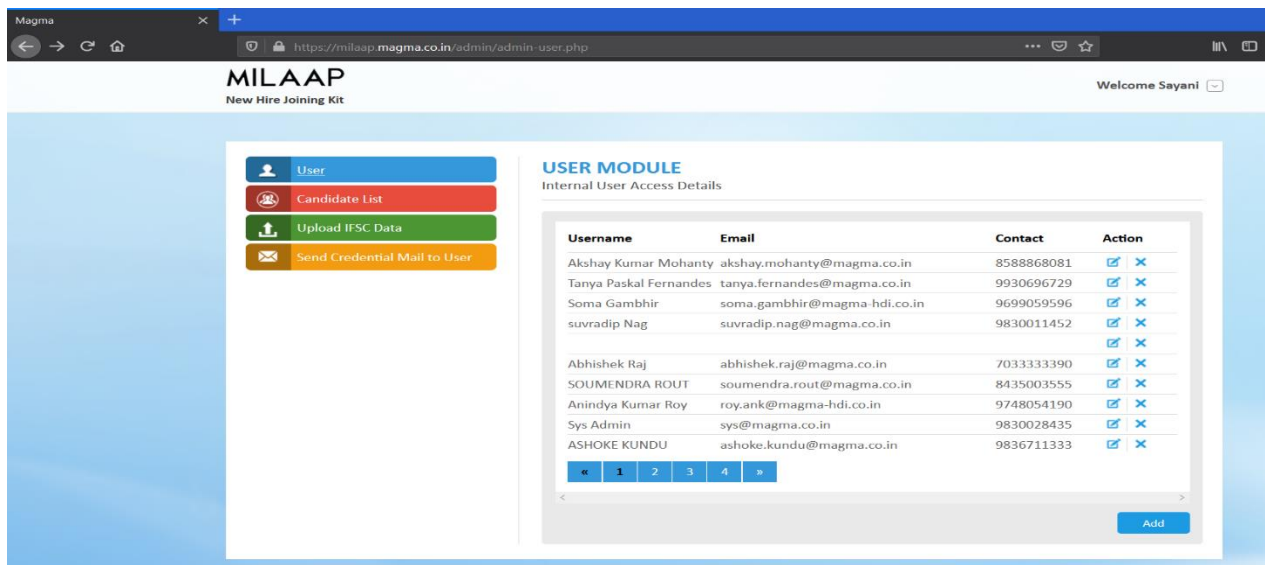
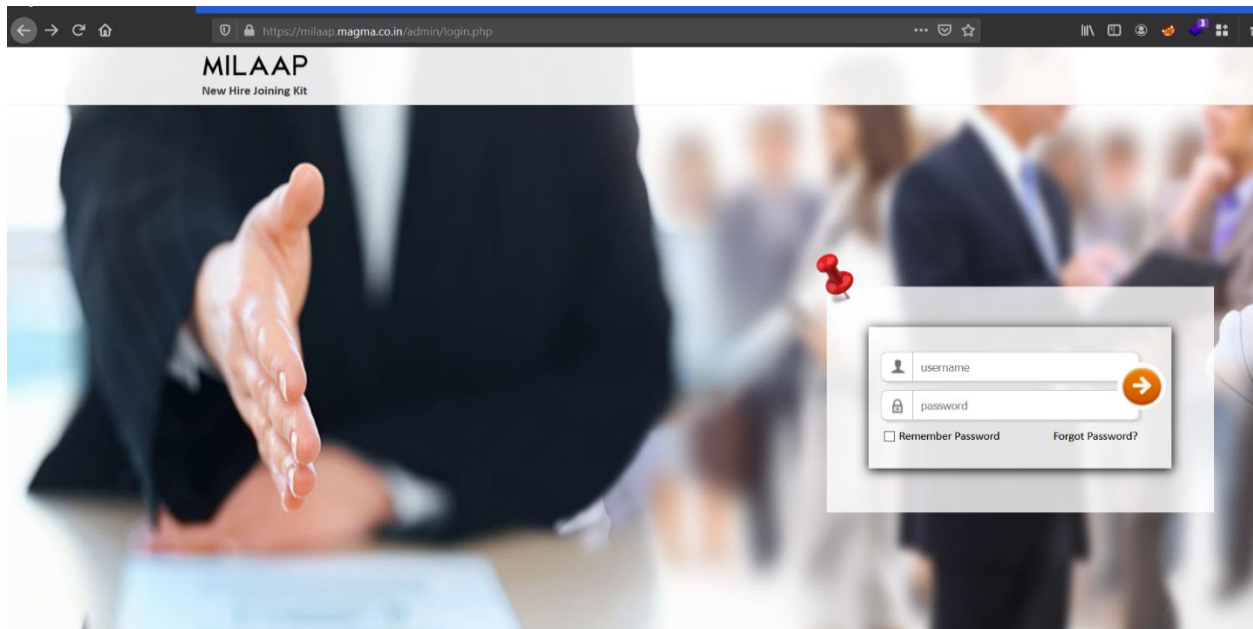


The screenshot shows a web browser window with the URL `https://milaap.magma.co.in/admin/admin-user.php`. The page title is "MILAAP New Hire Joining Kit" and it says "Welcome Sayani". On the left, there is a sidebar with four buttons: "User" (blue), "Candidate List" (red), "Upload IFSC Data" (green), and "Send Credential Mail to User" (orange). The main content area is titled "USER MODULE" and "Internal User Access Details". It contains a table with the following data:

Username	Email	Contact	Action
Akshay Kumar Mohanty	akshay.mohanty@magma.co.in	8588868081	✍ ✕
Tanya Paskal Fernandes	tanya.fernandes@magma.co.in	9930696729	✍ ✕
Soma Gambhir	soma.gambhir@magma-hdi.co.in	9699059596	✍ ✕
suvradip Nag	suvradip.nag@magma.co.in	9830011452	✍ ✕
Abhishek Raj	abhishek.raj@magma.co.in	7033333390	✍ ✕
SOUMENDRA ROUT	soumendra.rout@magma.co.in	8435003555	✍ ✕
Anindya Kumar Roy	roy.ank@magma-hdi.co.in	9748054190	✍ ✕
Sys Admin	sys@magma.co.in	9830028435	✍ ✕
ASHOKE KUNDU	ashoke.kundu@magma.co.in	9836711333	✍ ✕

Below the table is a pagination bar with buttons: « 1 2 3 4 » and a "Add" button at the bottom right.

Step2: After pressing back button, you will be prompted with login page. Now refresh the page and press forward. You will be logged back in. A redirection should be implemented after login request.



Solution: Implement URL redirection after processing login request.

5. Application has auto complete feature enabled

MEDIUM RISK

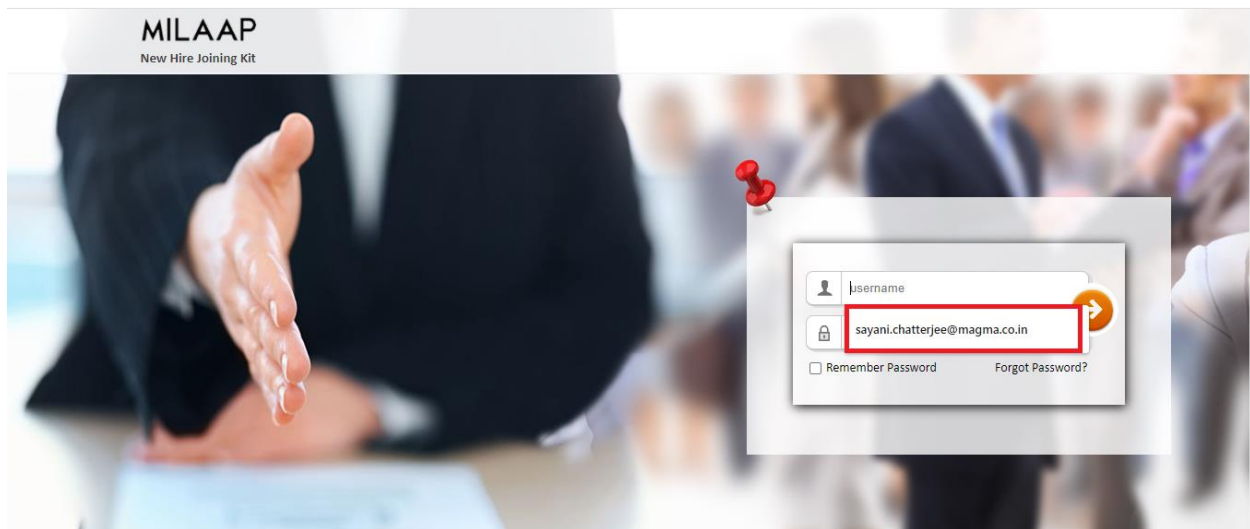
Description:

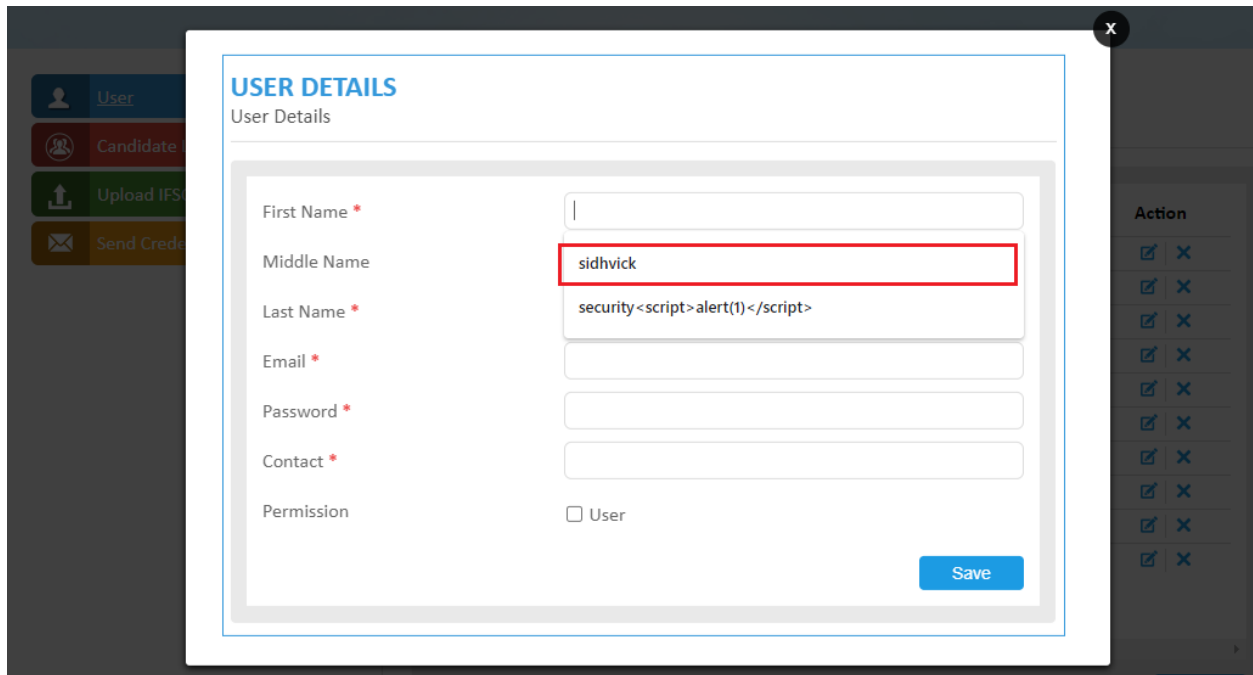
It was observed that full name, contact number, email Id, username and password field has autocomplete feature enabled. When new username and password is entered, and the form is submitted, the browser asks if the password should be saved. If user opts to save the password, then browser will save the credentials and prefill them on the login page. An attacker who has access to the victim's browser can login into the legitimate user's session by taking an advantage of this feature.

Vulnerable Location:

- Admin Login - <https://milaap.magma.co.in/admin/login.php>
- User Login - <https://milaap.magma.co.in/login.php>
- Add User - <https://milaap.magma.co.in/admin/admin-user.php>

Proof of Concept:





USER DETAILS
User Details

First Name *

Middle Name

Last Name *

Email *

Password *

Contact *

Permission

☐ User

Save

Solution:

Disable the autocomplete feature for username and password fields.

6. The application does not implement required session cookies attribute

Low

Description:

Part 1:

The "secure" cookie attribute instructs web browsers to only send the cookie through an encrypted HTTPS (SSL/TLS) connection. Here, the application does not implement "secure" cookie attribute. Therefore, an attacker can intercept session ID of legitimate user in clear text.

Part2:

The "HttpOnly" cookie attribute instructs web browsers not to allow scripts like JavaScript or VBscript from accessing session ID of the user. As this attribute is not enabled/set for session cookie in the application, an attacker can steal session ID through XSS attack.

If the HttpOnly attribute is set on a cookie, then the cookie's value cannot be read or set by client-side JavaScript. This measure makes certain client-side attacks, such as cross-site scripting, slightly harder to exploit by preventing them from trivially capturing the cookie's value via an injected script.

Vulnerable Location:

Throughout the application

- <https://Milaap.magma.co.in>

Proof of Concept:

```
1 HTTP/1.1 200 OK
2 Date: Tue, 02 Mar 2021 13:37:23 GMT
3 Set-Cookie: PHPSESSID=9rho64688vvpf7nhs76vrvi6ml; path=/
4 Expires: Thu, 19 Nov 1981 08:52:00 GMT
5 Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
  pre-check=0
6 Pragma: no-cache
7 Content-Type: text/html; charset=UTF-8
8 Set-Cookie: BNES_PHPSESSID=
  wc02HQe0XAiRTEkP5oesGQOX0jgtHwsMcGdbzF2Ym6L/qMqk11/rxCHXZnrSAZUVWzN
  LXD5V0UXAdGlecGml0Q==; path=/
9 Connection: close
10 Content-Length: 27319
11
12 <script>window.location.href='login.php'</script>
13 <!DOCTYPE html>
14 <html dir="ltr" lang="en">
15   <head>
16     <meta charset="utf-8">
17     <meta http-equiv="X-UA-Compatible" content="IE=Edge" />
18     <meta name="viewport" content="width=device-width,
19   initial-scale=1" />
20     <meta name="viewport" content="width=device-width,
21   height=device-height, initial-scale=1, minimum-scale=1,
```

Solution:

- Implement "secure" cookie attribute.
- Implement "HttpOnly" cookie attribute.

Recommendations:

- You should be aware that the restrictions imposed by the HttpOnly flag can potentially be circumvented in some circumstances, and that numerous other serious attacks can be delivered by client-side script injection, aside from simple cookie stealing.
- You should set the secure HttpOnly flag by including this attribute within the relevant Set-cookie directive and required to add RequiresSSL=true in web config.

Reference:

- <https://cwe.mitre.org/data/definitions/16.html>
- <https://cwe.mitre.org/data/definitions/614.html>
- <https://www.owasp.org/index.php/HttpOnly>

7. Application using vulnerable libraries

LOW RISK

Description:

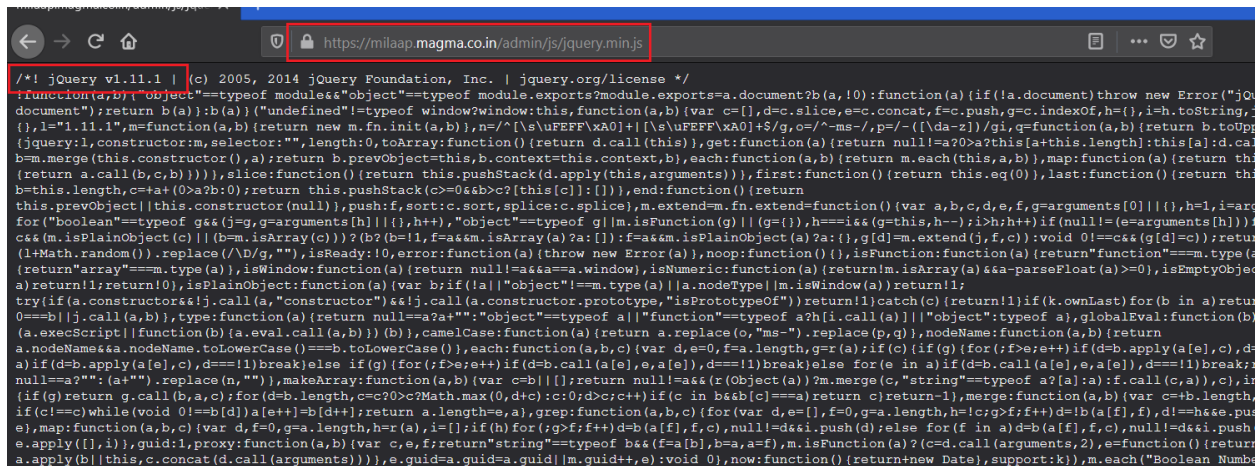
Libraries and frameworks used within the app almost always execute with full privileges. It is very common for web services to include a component with a known security vulnerability. The component with a known vulnerability could be the operating system itself, the CMS used, the web server, some plugin installed or even a library used by one of these plugins.

Vulnerable Location:

<https://milaap.magma.co.in/js>

Proof of Concept:

Case 1: It is observed that JQuery version is out of date.



Solution:

Update to latest version of jQuery with security patches.

8. Application is vulnerable to Clickjacking

LOW RISK

Description:

Clickjacking attack is an attack where an attacker uses multiple transparent layers to trick a user into clicking on a button or link on another page when the users were intending to click on the top-level page. The attackers load another page over a clickjacked page in a transparent layer. For instance, the hidden page may be an authentic page; therefore, the attackers can trick users into performing actions which the users never intended. Here, it was observed Clickjacking was possible when Pre-Login and Post Login if you're authenticated.

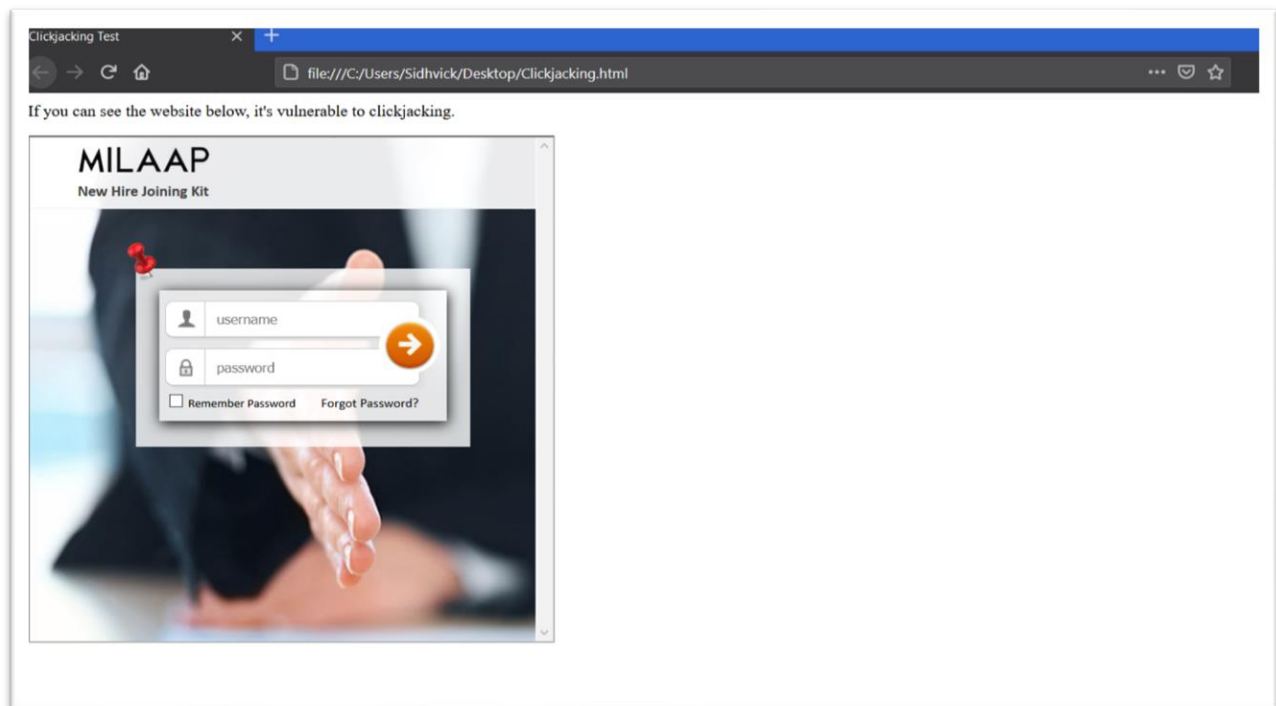
.

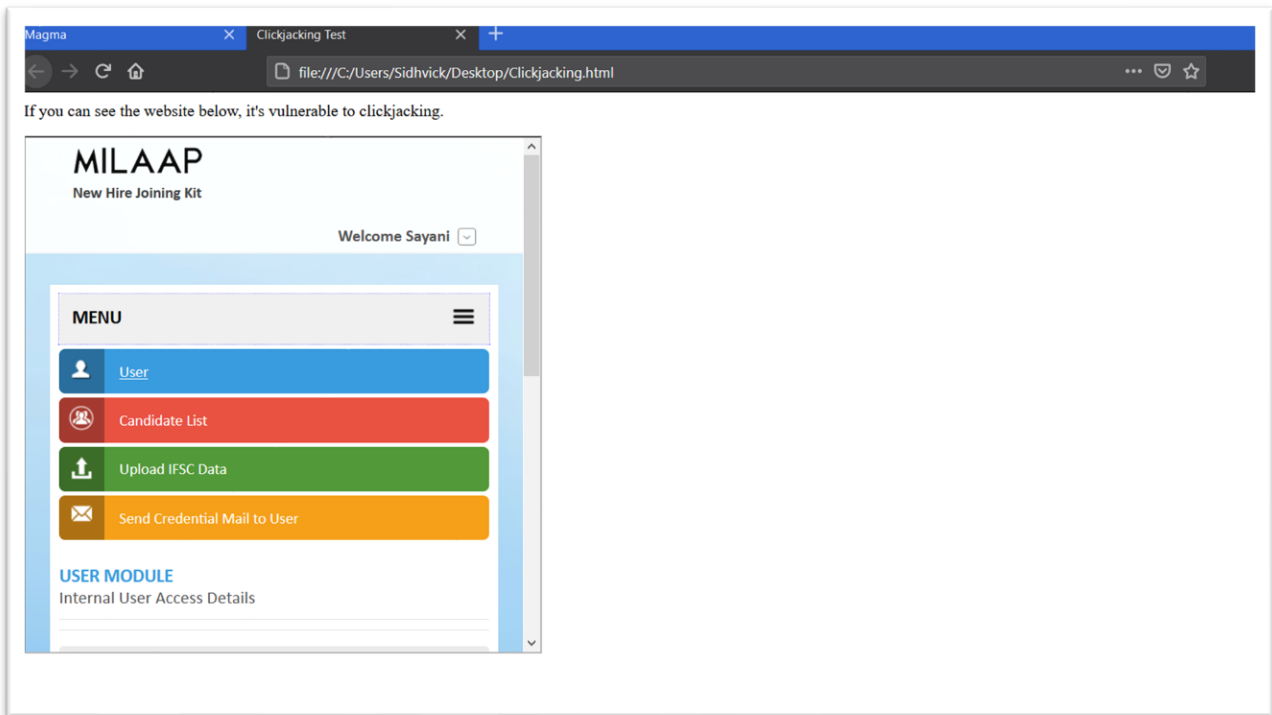
Vulnerable Location:

Throughout the application

<https://milaap.magma.co.in/>

POC:





Solution:

There are two main solutions to prevent Clickjacking which are as follow:

- Sending the X-Frame-Options HTTP response headers that instruct the browser to not allow framing from other domains

Employing frame bursting code in the UI to ensure that the current frame is the most top-level window.