Cross Site Scripting Vulnerability

Domain : vulnweb.com
Sub-Domain : testasp.vulnweb.com

Vulnerable URL:
http://testasp.vulnweb.com/Search.asp?tfSearch=%3Cscript%3Ealert%281%29%3C%2Fscript%3E

http://testasp.vulnweb.com/Search.asp?tfSearch=%3C%2Ftitle%3E%3Cscript%3Ealert%28%2FXSSPOSED%2F%29%3C%2Fscript%3E

HTTP POST data:
<script>alert(1)</script>
</title><script>alert(/xssposed/)</script>

Issue detail :

The name of an arbitrarily supplied URL parameter is copied into the HTML document as plain text between tags. The payload **<script>alert(1)</script>** was submitted in the name of an arbitrarily supplied URL parameter. This input was echoed unmodified in the application's response. This behavior demonstrates that it is possible to inject new HTML tags into the returned document. An attempt was made to identify a full proof-of-concept attack for injecting arbitrary JavaScript but this was not successful. You should manually examine the application's behavior and attempt to identify any unusual input validation or other obstacles that may be in place.

Issue background :

When data is unsafely duplicated from a request and echoed into the application's instant response, reflected cross-site scripting vulnerabilities result. An attacker can create a request using the vulnerability that, if sent by another application user, will allow the attacker's JavaScript code to run in the user's browser during that user's session with the application. The attacker-supplied malware can carry out a broad range of operations, including keylogging the victim's keystrokes, stealing their session token or login credentials, and carrying out arbitrary operations on the victim's behalf.

There are several approaches to persuade users to submit the forged request created by the attacker. In an email or instant message, for instance, the attacker may send the victim a link that leads to a malicious website. They can provide the URL to well-known websites that permit content creation, for instance in blog comments. Additionally, they have the ability to design a seemingly innocent website that would force anybody visiting it to send arbitrary cross-domain requests to the vulnerable application (using either the GET or the POST method).

The nature of the susceptible program, the types of data and functionality it includes, and the other apps that are part of the same domain and organization all affect how cross-site scripting vulnerabilities affect security. A cross-site scripting issue could be regarded as low risk if the program is simply used to show non-sensitive public material and has no login or access control features. The vulnerability may be leveraged to target other, more security-critical apps, too, therefore it might be deemed high risk if the same application is located on a domain that can access cookies for other applications. The vulnerability could also be used to give credibility to phishing attacks if the company that owns the application is a likely target for such attacks by injecting Trojan functionality into the vulnerable application and taking advantage of users' trust in the company to steal login information for other applications that the company owns. Cross-site scripting should always be regarded as high risk in many different sorts of applications, such as those that provide online banking functions.

Issue remediation :

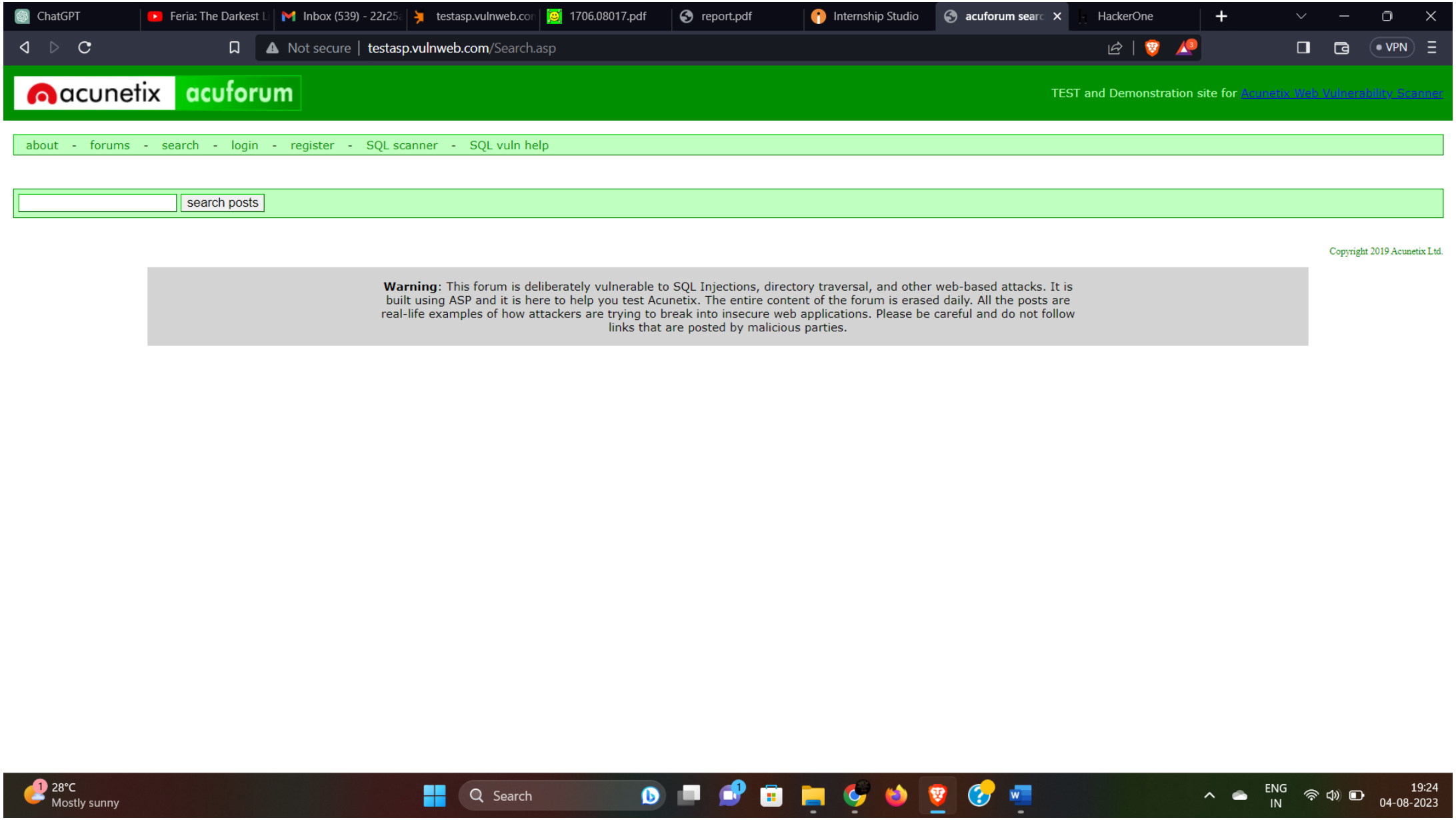
Cross-site scripting attacks can often be stopped using two levels of security when user-controllable data is transferred into application responses:Given the intended content, input should be vetted as precisely as feasible when it arrives.to encompass. Personal names, for instance, should be composed of a limited number of alphabeticaltypographic characters, but also being brief; Four days should make up a birth year.Email addresses should fit a well specified regular expression and contain numbers. Information that fails theNot sanitized, but rejected, should be validation.At each point when user input is copied into application answers, it should be HTML-encoded. AllHTML metacharacters like > "'and = should be changed to their corresponding HTML characters.entities (such as >).

It is necessary to parse the supplied HTML to ensure that it does not use any risky syntax when the application's functionality permits users to author content using a constrained subset of HTML tags and attributes (for example, blog comments that allow limited formatting and linking).

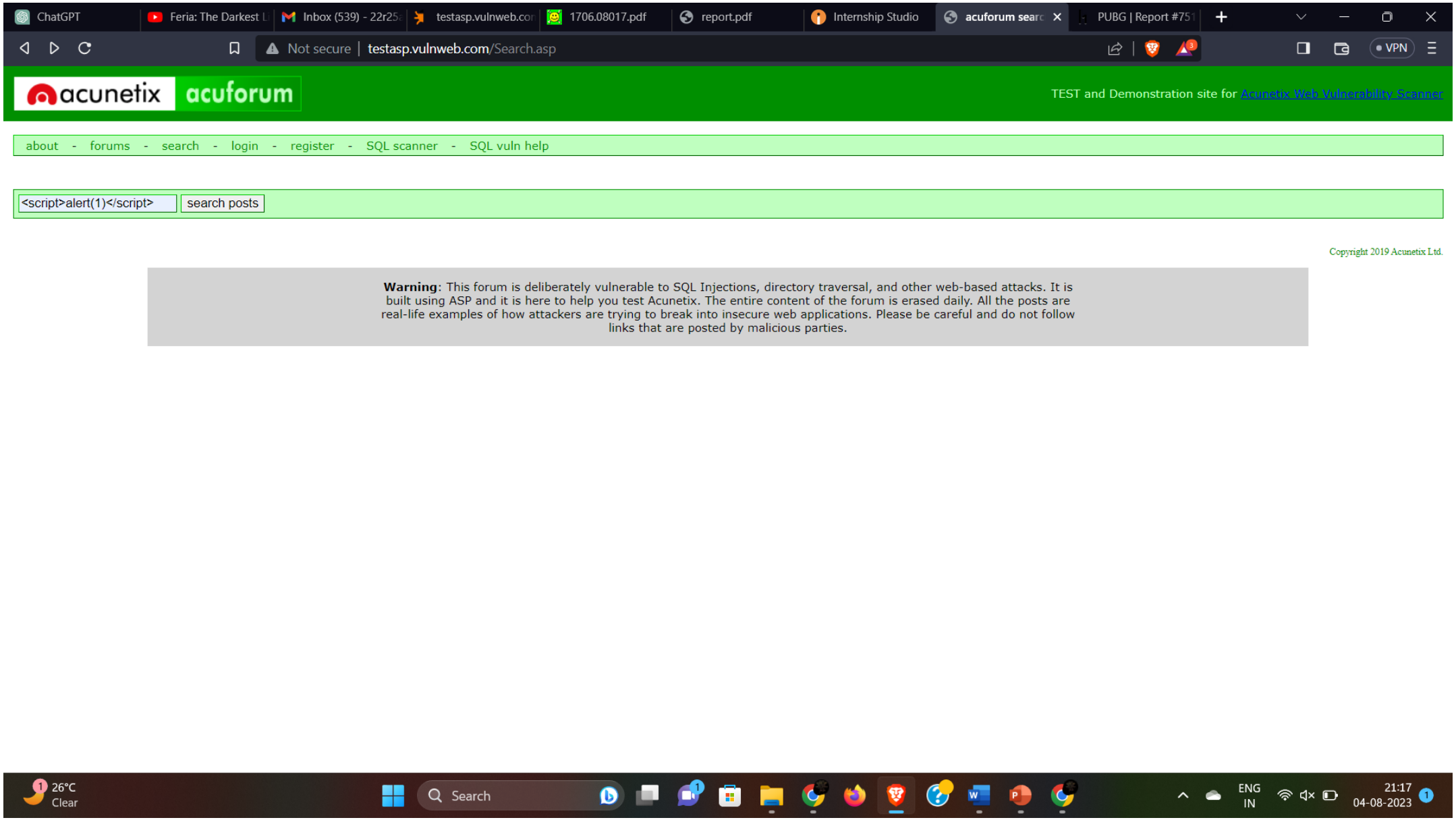
References

- Cross-site scripting
- Reflected cross-site scripting
- Using Burp to Find XSS issues

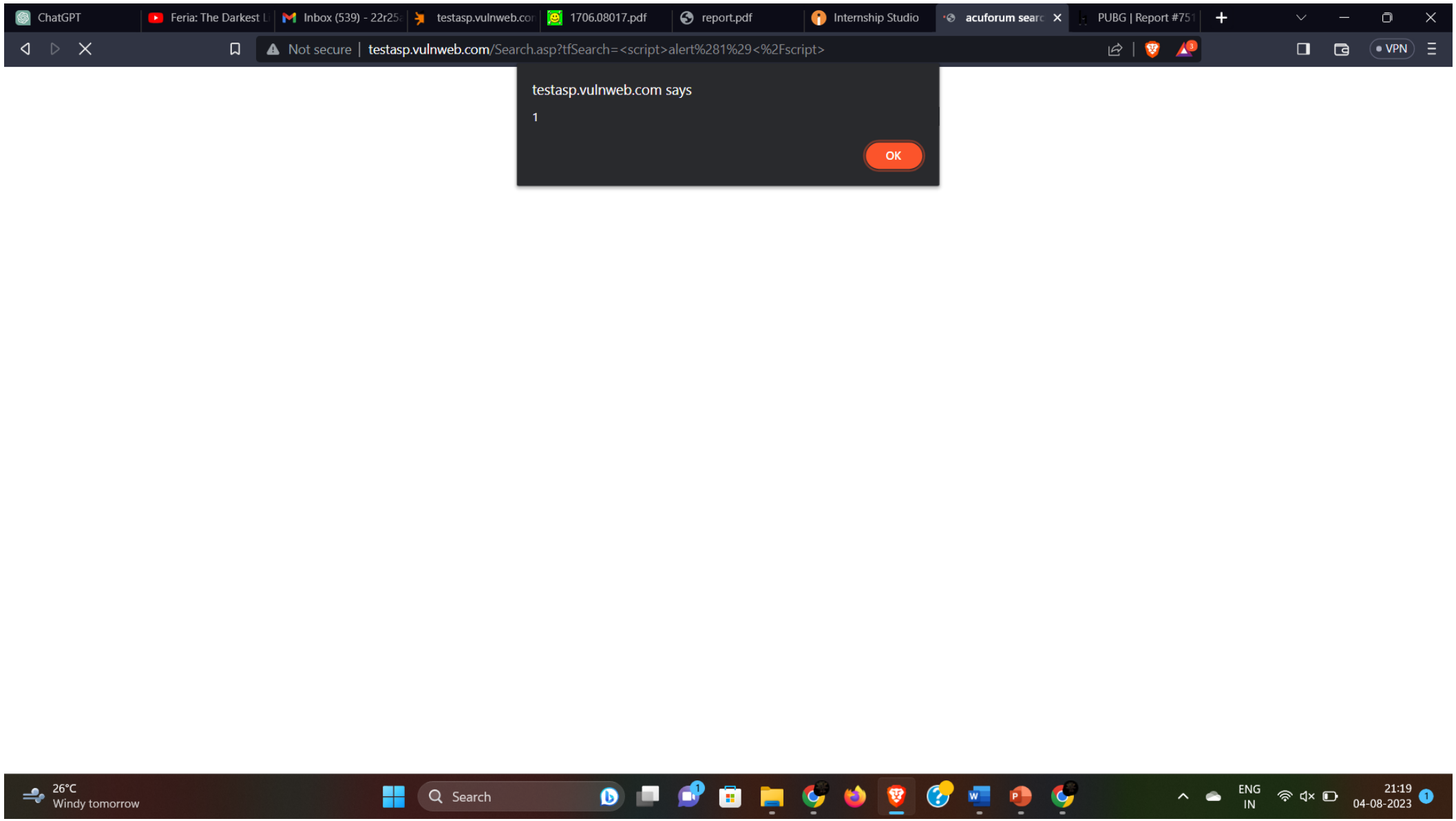
STEP 1 :


ChatGPT
Feria: The Darkest L
Inbox (539) - 22r25
testasp.vulnweb.com
1706.08017.pdf
report.pdf
Internship Studio
acuforum searc
HackerOne
Not secure | testasp.vulnweb.com/Search.asp
VPN
acunetix
acuforum
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
about - forums - search - login - register - SQL scanner - SQL vuln help
search posts
Copyright 2019 Acunetix Ltd.
Warning: This forum is deliberately vulnerable to SQL Injections, directory traversal, and other web-based attacks. It is built using ASP and it is here to help you test Acunetix. The entire content of the forum is erased daily. All the posts are real-life examples of how attackers are trying to break into insecure web applications. Please be careful and do not follow links that are posted by malicious parties.
28°C
Mostly sunny
Search
ENG
IN
19:24
04-08-2023

STEP 2 :


ChatGPT
Feria: The Darkest L
Inbox (539) - 22r25
testasp.vulnweb.com
1706.08017.pdf
report.pdf
Internship Studio
acuforum searc
PUBG | Report #751
Not secure | testasp.vulnweb.com/Search.asp
VPN
acunetix
acuforum
TEST and Demonstration site for Acunetix Web Vulnerability Scanner
about - forums - search - login - register - SQL scanner - SQL vuln help
<script>alert(1)</script>
search posts
Copyright 2019 Acunetix Ltd.
Warning: This forum is deliberately vulnerable to SQL Injections, directory traversal, and other web-based attacks. It is built using ASP and it is here to help you test Acunetix. The entire content of the forum is erased daily. All the posts are real-life examples of how attackers are trying to break into insecure web applications. Please be careful and do not follow links that are posted by malicious parties.
26°C
Clear
Search
ENG
IN
21:17
04-08-2023

STEP 3 :


ChatGPT
Feria: The Darkest L
Inbox (539) - 22r25
testasp.vulnweb.com
1706.08017.pdf
report.pdf
Internship Studio
acuforum searc
PUBG | Report #751
Not secure | testasp.vulnweb.com/Search.asp?tfSearch=<script>alert%281%29<%2Fscript>
VPN
testasp.vulnweb.com says
1
OK
26°C
Windy tomorrow
Search
ENG
IN
21:19
04-08-2023

PROCEDURE VIDEO :