

Sri Lanka Institute of Information Technology



BUG BOUNTY REPORT - 7

Web Security – IE2062

IT22362780

Jayaweera N.S

Report Details

Report # - 07

Domain - <https://pinterest.com>

Platform -bugcrowd.com

Scans performed - Recon-ng scan
Nmap scan
Wafw00f scan
Dotdotpwn scan
Nikto scan
CSRF scan
Manual scanning using Wapplyzer
Text injection testing
File upload vulnerability testing
Command injection testing
XSS injection
Nslookup scan
Sqlmap scan

Nmap scan

Using nmap scan all the open ports in the target can be identified.

```
(kali@kali)-[~]
$ nmap -sS -T4 pinterest.com
You requested a scan type which requires root privileges.
QUITTING!

(kali@kali)-[~]
$ sudo nmap -sS -T4 pinterest.com
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-27 16:25 EDT
Nmap scan report for pinterest.com (151.101.64.84)
Host is up (0.012s latency).
Other addresses for pinterest.com (not scanned): 151.101.128.84 151.101.0.84 151.101.192.84
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 10.26 seconds
```

No unusual ports are open.

But SmtP port 25 is vulnerable when it's opened, because it lacks authentication and encryption.

Let's see if we can establish a connection on port 25.

```
(kali@kali)-[~]
$ nmap pinterest.com --script=smtp* -p 25
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-27 16:26 EDT
Nmap scan report for pinterest.com (151.101.128.84)
Host is up (0.051s latency).
Other addresses for pinterest.com (not scanned): 151.101.0.84 151.101.64.84 151.101.192.84

PORT      STATE SERVICE
25/tcp    open  smtp
|_smtp-open-relay: Couldn't establish connection on port 25
|_smtp-enum-users:
|_ Couldn't establish connection on port 25
|_smtp-commands: Couldn't establish connection on port 25
```

Connection cannot be established therefore a vulnerability cannot be identified.

Recon-ng

here the recon-ng will be used to find all the sub domains in the target.

```
[recon-ng][default] > workspaces create bb1
[recon-ng][bb1] > modules load hackertarget
[recon-ng][bb1][hackertarget] > show options
Shows various framework items

Usage: show <companies|contacts|credentials|domains|hosts|leaks|locations|netblocks|ports|profiles|p

[recon-ng][bb1][hackertarget] > set SOURCE pinterest.com
[!] Invalid command: set SOURCE pinterest.com.
[recon-ng][bb1][hackertarget] > options set SOURCE pinterest.com
SOURCE ⇒ pinterest.com
[recon-ng][bb1][hackertarget] > run
```

PINTEREST.COM

```
[*] Country: None
[*] Host: crawl-54-236-1-10.pinterest.com
[*] Ip_Address: 54.236.1.10
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -----
[*] Country: None
[*] Host: crawl-54-236-1-101.pinterest.com
[*] Ip_Address: 54.236.1.101
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -----
[*] Country: None
[*] Host: crawl-54-236-1-102.pinterest.com
[*] Ip_Address: 54.236.1.102
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*] -----
[*] Country: None
[*] Host: crawl-54-236-1-104.pinterest.com
[*] Ip_Address: 54.236.1.104
```

SUMMARY

```
[*] 196 total (196 new) hosts found.
[recon-ng][bb1][hackertarget] > █
```

196 total subdomains were found.

Nslookup


```
(kali㉿kali)-[~]  
$ nslookup pinterest.com  
Server:      192.168.1.1  
Address:     192.168.1.1#53  
  
Non-authoritative answer:  
Name:   pinterest.com  
Address: 151.101.128.84  
Name:   pinterest.com  
Address: 151.101.192.84  
Name:   pinterest.com  
Address: 151.101.0.84  
Name:   pinterest.com  
Address: 151.101.64.84
```

The ip address of the web application is found.

Wafw00f scan

Used to identify the type of WAF that is used to protect the web application.

```
(kali@kali)-[~]
$ wafw00f https://pinterest.com
```



The logo features a central blue horizontal bar with a green outline, flanked by orange and green parentheses. Above it is a white hexagon with the word "Woof!" in black. To the right is a vertical stack of red and orange parentheses.

```
~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit

[*] Checking https://pinterest.com
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

No WAF detected.

Wapplyzer

The Wapplyzer is used to identify the technologies used in the web application.

The screenshot displays the Wappalyzer web application interface. At the top, there is a purple header with the Wappalyzer logo and navigation icons. Below the header, there are tabs for 'TECHNOLOGIES' and 'MORE INFO', and an 'Export' button. The main content area is divided into two columns. The left column lists various technologies under different categories: JavaScript frameworks (React), Security (HSTS, reCAPTCHA), Miscellaneous (HTTP/3, Module Federation with 50% sure, Open Graph), Reverse proxies (Module Federation with 50% sure, Open Graph), PWA (PWA), Webpack (50% sure), CDN (Amazon S3), and Advertising (LinkedIn Ads). The right column lists JavaScript libraries (core-js 3.8.3, fullPage.js 3.1.0, Loadable-Components, JQuery), PaaS (Amazon Web Services), Reverse proxies (Envoy), Authentication (Google Sign-in), and Performance (Priority Hints). Each technology is represented by a circular icon and a link to its official website.

Wappalyzer

TECHNOLOGIES MORE INFO Export

JavaScript frameworks

- React

JavaScript libraries

- core-js 3.8.3
- fullPage.js 3.1.0
- Loadable-Components
- JQuery

Security

- HSTS
- reCAPTCHA

Miscellaneous

- HTTP/3
- Module Federation 50% sure
- Open Graph

PaaS

- Amazon Web Services

Reverse proxies

- Envoy

Reverse proxies

- Envoy

Authentication

- Google Sign-in

Performance

- Priority Hints

CDN

- Amazon S3

Advertising

- LinkedIn Ads

Something wrong or missing?

These are the technologies used.

core-js@3.8.3 vulnerabilities

Standard library

Direct Vulnerabilities

No direct vulnerabilities have been found for this package in Snyk's vulnerability database. This does not include vulnerabilities belonging to this package's dependencies.

There is no direct vulnerability in the core.js

fullpage-react@3.1.0 vulnerabilities

Stateful fullpage.js inspired scrolling for React

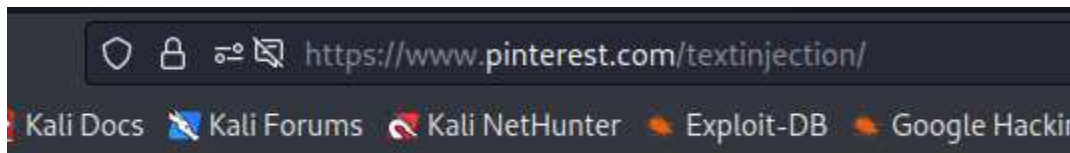
Direct Vulnerabilities

No direct vulnerabilities have been found for this package in Snyk's vulnerability database. This does not include vulnerabilities belonging to this package's dependencies.

And there is no direct vulnerability in the fullpage.js too.

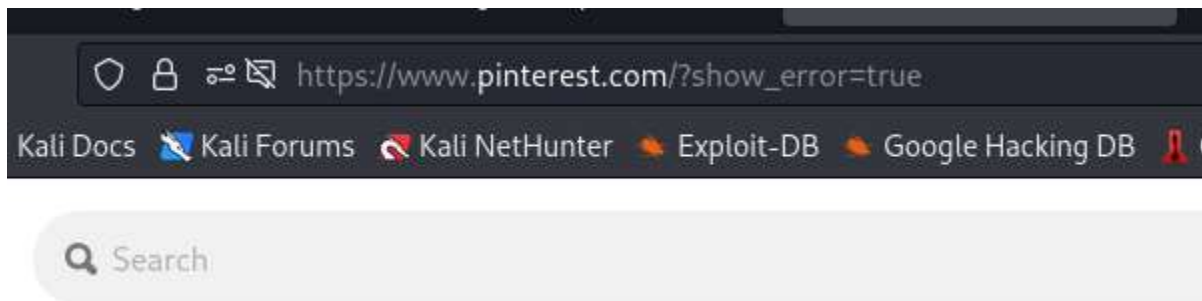
Text injection

An arbitrary string value is appended to the URL to see whether the web application is vulnerable towards a text injection.



If the entered text is reflected on the error response of the web page, there is a possibility to inject malicious content.

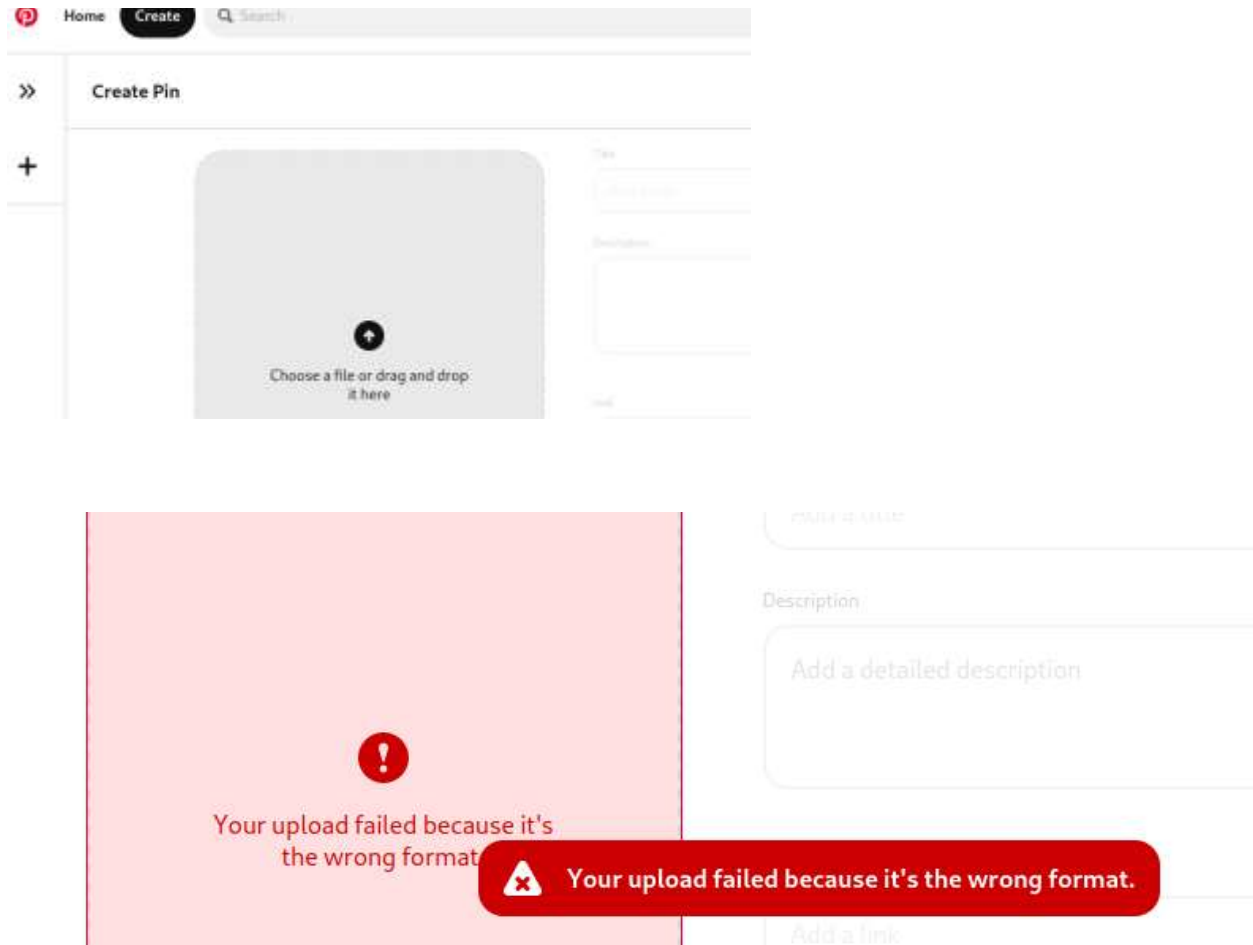
If not, the web application is safe.



No text injection vulnerability can be found.

File upload vulnerability

If a .php file can be uploaded from the file uploading facility, there is a possibility to upload and execute a reverse shell php code.



No vulnerability found.

Dotdotpwn

Dotdotpwn is a directory traversal checker.

```
[+] Report name: Reports/pinterest.com_04-27-2024_16-44.txt

[===== TARGET INFORMATION =====]
[+] Hostname: pinterest.com
[+] Protocol: http
[+] Port: 80

[===== TRAVERSAL ENGINE =====]
[+] Creating Traversal patterns (mix of dots and slashes)
[+] Multiplying 6 times the traversal patterns (-d switch)
[+] Creating the Special Traversal patterns
[+] Translating (back)slashes in the filenames
[+] Adapting the filenames according to the OS type detected (unix)
[+] Including Special suffixes
[+] Traversal Engine DONE ! - Total traversal tests created: 11028

[===== TESTING RESULTS =====]
[+] Ready to launch 3.33 traversals per second
[+] Press Enter to start the testing (You can stop it pressing Ctrl + C)

[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/issue ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
```

```
[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/issue ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/issue ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/issue ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/issue ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/passwd ← VULNERABLE!
[*] Testing Path: http://pinterest.com:80/../../../../etc/issue ← VULNERABLE!
```

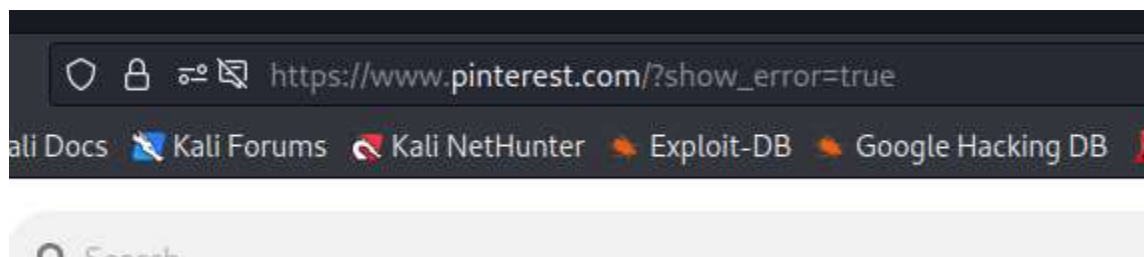
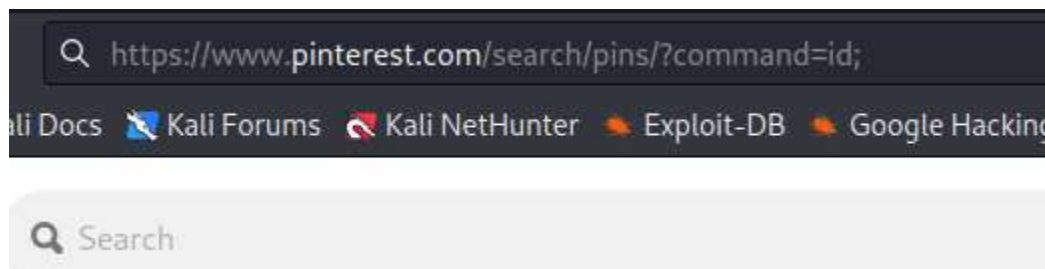
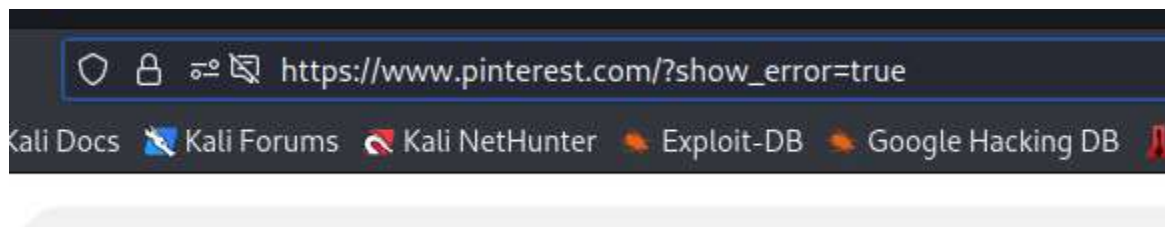
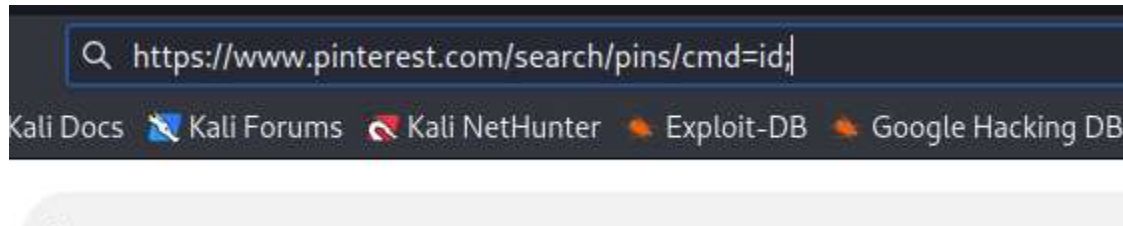
The scan results returned status codes within the range 400 (400-499). It shows a client error.

There are some URLs vulnerable to directory traversal.

Command injection

The query that is used for searching is used against this vulnerability.

The “id” command is appended to the url.



No command injection vulnerability can be found.

CSRF scan

Testing csrf on the change password feature

Change your password

Old password · [Forgot it?](#)

.....

✖ Your old password was entered incorrectly. Please enter it again.

New password

.....

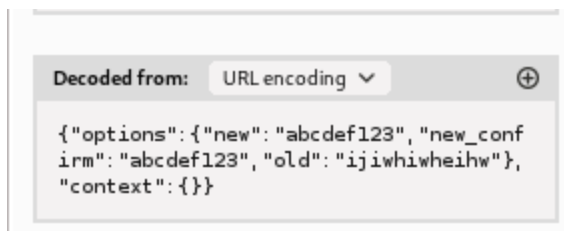
Type it again

.....

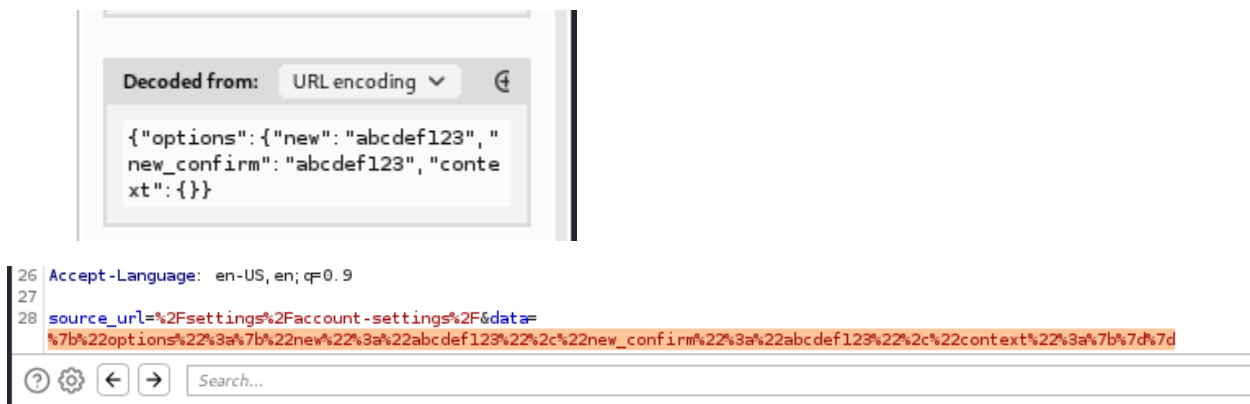
Cancel Change Password

Capture the request using burpsuite. The data is url encoded so it needs to be decoded.

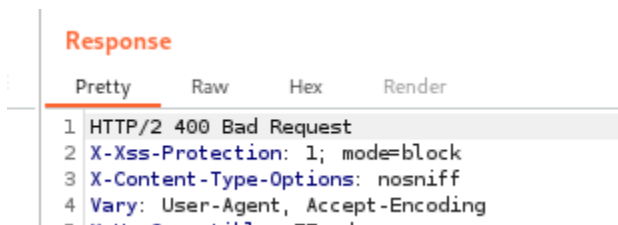
```
23 Sec-Fetch-Dest: empty
24 Referer: https://www.pinterest.com/
25 Accept-Encoding: gzip, deflate
26 Accept-Language: en-US,en;q=0.9
27
28 source_url=%2Fsettings%2Faccount-settings%2F&data=
%7B%22options%22%3A%7B%22new%22%3A%22abcdef123%22%2C%22new_confirm%22%3A%22abcdef
123%22%2C%22old%22%3A%22ijihwheihw%22%7D%2C%22context%22%3A%7B%7D%7D
```



Remove the old password from the field and check if it is being validated or not.



AS it can be seen below, once we remove the old password and send the data, a bad request response is received. Hence a csrf is not possible as the old password is validated.



Sqlmap

With the use of this scan, we can identify whether a sql injection can be done or not.

```
(kali㉿kali)-[~]
$ sqlmap -u https://www.pinterest.com/search/pins/?q=hello --databases=mysql --tables=mysql.* --level=0 --risk=0 --tamper=random-agent --random-agent
{1.6.11#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to abide by the applicable laws of their country/region. Developers assume no liability and are not responsible for any misuse or damage caused by your actions.

[*] starting @ 03:04:35 /2024-04-30/

[03:04:36] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('_auth=0;_pinterest_sess=TWc9PSZuYXp...bnhjJTGp=b7eb8fb5a2f...47e230ae0'). Do you want to use those [Y/n] y
[03:05:02] [INFO] checking if the target is protected by some kind of WAF/IPShield
[03:05:05] [WARNING] reflective value(s) found and filtering out
[03:05:05] [INFO] testing if the target URL content is stable
[03:05:08] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page comparison on parameters are detected, or in case of junk results, refer to user's manual paragraph 'Page comparison'
how do you want to proceed? [(C)ontinue/(S)tring/(R)egex/(Q)uit] c
[03:05:17] [INFO] testing if GET parameter 'q' is dynamic
[03:05:21] [WARNING] GET parameter 'q' does not appear to be dynamic

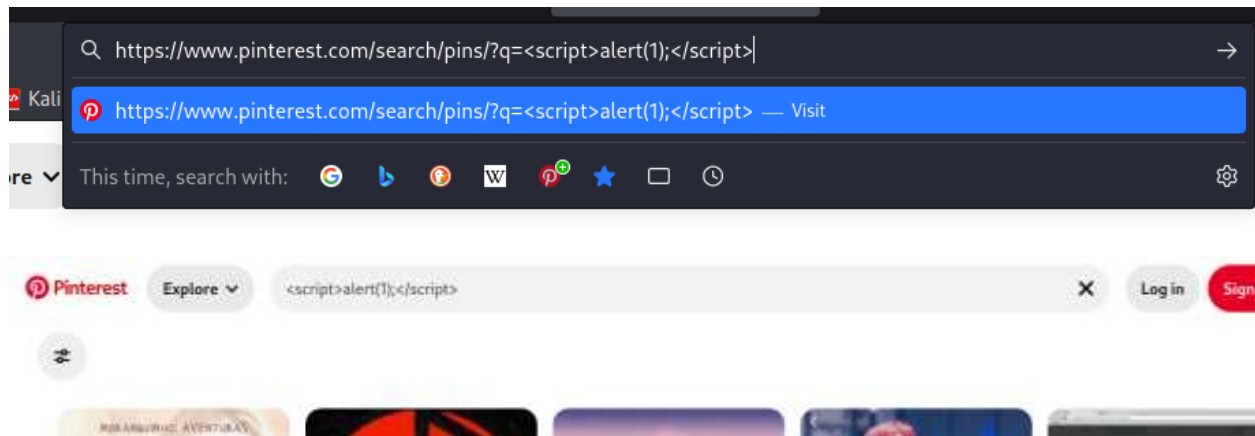
[03:07:19] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[03:07:27] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[03:07:41] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to continue? [Y/n] y
[03:08:48] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[03:09:08] [WARNING] GET parameter 'q' does not seem to be injectable
[03:09:08] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level/--risk' options if you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper=random-agent'
[03:09:08] [WARNING] your sqlmap version is outdated

[*] ending @ 03:09:08 /2024-04-30/
```

There is no injection vulnerability in the above web application.

XSS vulnerability

A payload is appended to the url to test against xss injection.



It is treated as a text instead of a script.

Therefore, it's not vulnerable to XSS injection.