

Sri Lanka Institute of Information Technology



BUG BOUNTY REPORT - 10

Web Security – IE2062

IT22362780

Jayaweera N.S

Report Details

Report # - 10

Domain - <https://soundcloud.com>

Platform -bugcrowd.com

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

Nmap scan

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

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS -T4 soundcloud.com
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-02 11:53 EDT
Nmap scan report for soundcloud.com (13.33.88.55)
Host is up (0.014s latency).
Other addresses for soundcloud.com (not scanned): 13.33.88.75 13.33.88.80 13.33.88.85
rDNS record for 13.33.88.55: server-13-33-88-55.sin2.r.cloudfront.net
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 5.58 seconds
```

No unusual ports are open.

But Smtip 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 using Metasploit tool.

Nikto scan

```

Certificate: TLS_AES128_GCM_SHA256
Issuer: /C=BE/O=GlobalSign nv-ba/CN=GlobalSign GCC R3 DV TLS CA 2020
+ Message: Multiple IP addresses found: 13.33.88.80, 13.33.88.55, 13.33.88.85, 13.33.88.75
+ Start Time: 2024-05-02 11:50:08 (GMT-4)

+ Server: am/2
+ Retrieved via header: 1.1 f378d87611123aa47c006262522a5a04.cloudfront.net (CloudFront)
+ Uncommon header "x-pants" found, with contents: distant-towel
+ Uncommon header "x-cache" found, with contents: Miss from cloudfront
+ Uncommon header "x-amz-cf-id" found, with contents: 3an88hb_0S1-uydWXC5uhsf01ooqlA_uFZP5wI0C_h_lC1fdDesXew=
+ Uncommon header "x-amz-cf-pop" found, with contents: SIN2-P2
+ Uncommon header "server-timing" found, with contents: enabledFeatures; dur=8.360849; desc="api-v2/enabledFeatures"; experiments; dur=11.305593; desc="api-v2/privacySettings"; geoip; dur=1.544445; desc="geoip/geoip"; privacySettings; dur=5.727589; desc="api-v2/privacySettings"
+ The site uses SSL and Expect-CT header is not present.
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ All CGI directories "found", use "-C none" to test none
+ /crossdomain.xml contains 2 lines which include the following domains: *.soundcloud.com *.sndcdn.com
+ Uncommon header "x-request-id" found, with contents: 6633B9ED796880A5C7D3
+ Server banner has changed from 'am/2' to 'CloudFront' which may suggest a WAF, load balancer or proxy is in place
+ The Content-Encoding header is set to 'deflate' this may mean that the server is vulnerable to the BREACH attack.
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect: SSL negotiation failed: error:0A000410:SSL routines::ssl
shake failure at /var/lib/nikto/plugins/LM2.pm line 5157.
at /var/lib/nikto/plugins/LM2.pm line 5157.
; at /var/lib/nikto/plugins/LM2.pm line 5157.
+ Scan terminated: 20 error(s) and 11 item(s) reported on remote host
+ End Time: 2024-05-02 12:02:16 (GMT-4) (188 seconds)

+ 1 host(s) tested
```

Scan results:

- The site uses SSL and Expect-CT header is not present.

The "expect-CT" header is a security feature that helps websites, and their users avoid the risks associated with incorrectly issued SSL certificates.

It supports transparency and accountability when issuing SSL certificates, which improves overall web security.

There are some issues/disadvantages occurred when the “expect-CT” header is absent:

- The protection against the mis issuing of SSL certificates will be low.
- Mismanagement of SSL certificates.

- No trust and security

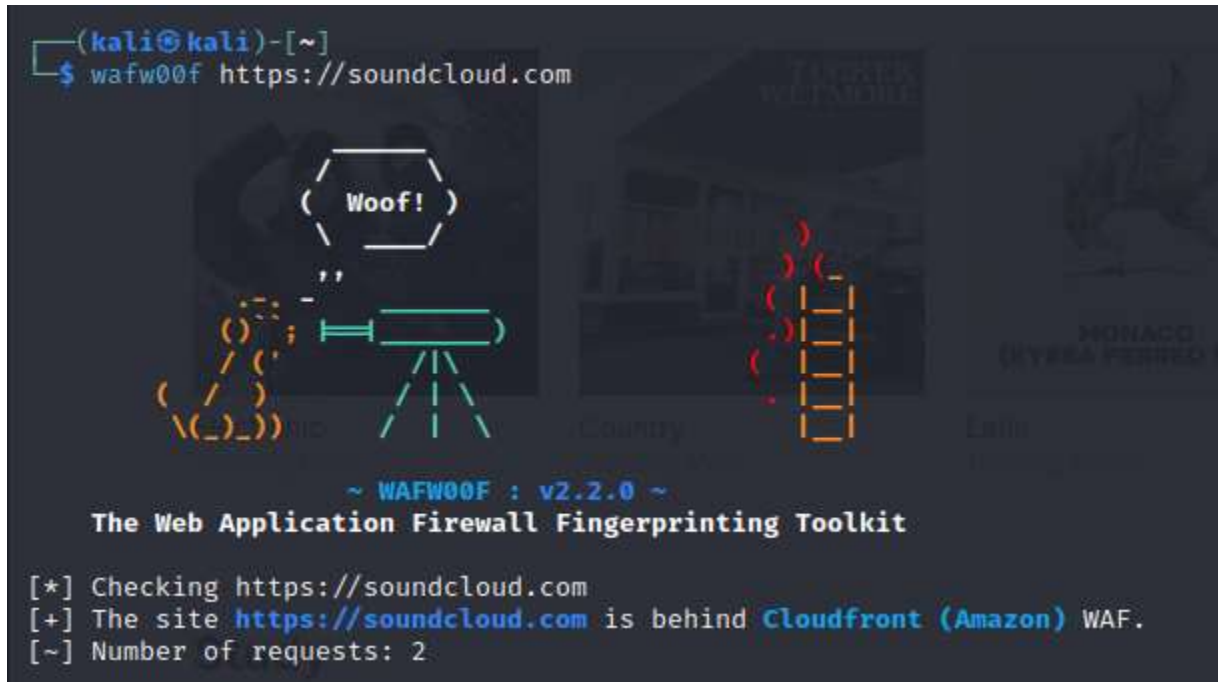
But the absence of “expected-CT” header is not a huge vulnerability or a security issue in a website.

- The X-content-Type-options header is not set.

Wafw00f

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

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



```
~ WAFW00F : v2.2.0 ~  
The Web Application Firewall Fingerprinting Toolkit  
[*] Checking https://soundcloud.com  
[+] The site https://soundcloud.com is behind Cloudfront (Amazon) WAF.  
[~] Number of requests: 2
```

The WAF used is Cloudfront.

Nslookup

```
(kali@kali)-[~]  
$ nslookup soundcloud.com  
Server:      192.168.8.1  
Address:     192.168.8.1#53  
  
Non-authoritative answer:  
Name:   soundcloud.com  
Address: 13.33.88.80  
Name:   soundcloud.com  
Address: 13.33.88.85  
Name:   soundcloud.com  
Address: 13.33.88.55  
Name:   soundcloud.com  
Address: 13.33.88.75
```

The relevant ip address of the domain is found.

Dotdotpwn

Dotdotpwn is a directory traversal checker.

```
[===== 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)


[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/../etc/issue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/../../etc/issue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/../../../../etc/passwd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/../../../../etc/issue

[*] HTTP Status: 500 | Testing Path: http://soundcloud.com:80/..%c1k1e%c1k1e%c1k1e%c1k1e%c1k1e%c1k1ecetcc1k1cpasswd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1e%c1k1e%c1k1e%c1k1e%c1k1e%c1k1ecetcc1k1ciissue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1afetc%c1k1afpasswd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1afetc%c1k1afiissue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1afetc%c1k1afpasswd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1afetc%c1k1afiissue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1af..%c1k1afetc%c1k1afpasswd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1af..%c1k1af..%c1k1afetc%c1k1afiissue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1af..%c1k1af..%c1k1afetc%c1k1afpasswd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1af..%c1k1af..%c1k1afetc%c1k1afiissue
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1af..%c1k1af..%c1k1af..%c1k1afetc%c1k1afpasswd
[*] HTTP Status: 400 | Testing Path: http://soundcloud.com:80/..%c1k1af..%c1k1af..%c1k1af..%c1k1af..%c1k1afetc%c1k1afiissue
```

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

Vulnerable paths were found.

Metasploit

```
msf6 > search smtp
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/smtp/apache_james_exec	2015-10-01	normal	Yes	Apache James Server
1	auxiliary/server/capture/smtp		normal	No	Authentication Cap
2	auxiliary/scanner/http/gavazzi_em_login_loot		normal	No	Carlo Gavazzi Ene
p Plant Database					
3	exploit/unix/smtp/clamav_milter_blackhole	2007-08-24	excellent	No	ClamAV Milter Bla
4	exploit/windows/browser/communicrypt_mail_activex	2010-05-19	great	No	CommuniCrypt Mail
5	exploit/linux/smtp/exim_ghostbyname_bof	2015-01-27	great	Yes	Exim GHOST (glibc
6	exploit/linux/smtp/exim4_dovecot_exec	2013-05-03	excellent	No	Exim and Dovecot
7	exploit/unix/smtp/exim4_string_format	2010-12-07	excellent	No	Exim4 string form
8	auxiliary/client/smtp/emailer		normal	No	Generic Emailer (\$
9	exploit/linux/smtp/haraka	2017-01-26	excellent	Yes	Haraka SMTP Comm
10	exploit/windows/http/mdaemon_worldclient_form2raw	2003-12-29	great	Yes	MDaemon WorldClie
11	exploit/windows/smtp/ms03_046_exchange2000_xexch50	2003-10-15	good	Yes	MS03-046 Exchange

Search for smtp.

Use the module “fuzzer” to fuzz the smtp service and use “smtp_enum” to enumerate the usernames.

```
msf6 auxiliary(fuzzers/smtp/smtp_fuzzer) > set RHOSTS soundcloud.com
RHOSTS => soundcloud.com
msf6 auxiliary(fuzzers/smtp/smtp_fuzzer) > run

[-] 13.33.88.85:25 - Connection reset by peer
[*] 13.33.88.85:25 - Fuzzing with iteration 1

[-] 13.33.88.85:25 - Connection reset by peer
[*] 13.33.88.85:25 - Fuzzing with iteration 2

[-] 13.33.88.85:25 - Connection reset by peer
[*] 13.33.88.85:25 - Fuzzing with iteration 3

[-] 13.33.88.85:25 - Connection reset by peer
[*] 13.33.88.85:25 - Fuzzing with iteration 4

[-] 13.33.88.75:25 - EOFError
[*] 13.33.88.75:25 - Fuzzing with iteration 97

[-] 13.33.88.75:25 - EOFError
[*] 13.33.88.75:25 - Fuzzing with iteration 98

[-] 13.33.88.75:25 - EOFError
[*] 13.33.88.75:25 - Fuzzing with iteration 99

[-] 13.33.88.75:25 - EOFError
[*] 13.33.88.75:25 - Fuzzing with iteration 100

[*] soundcloud.com:25 - Scanned 4 of 4 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(fuzzers/smtp/smtp_fuzzer) >
```

Fuzzing failed due to connection time out indicating inability to enumerate the service. Fuzzing attempts are blocked by the server.

Wapplyzer

The screenshot displays the Wappalyzer web application interface. At the top, there is a purple header with the Wappalyzer logo and name, along with a toggle switch, a settings gear icon, and a refresh icon. Below the header, there are two tabs: 'TECHNOLOGIES' (selected) and 'MORE INFO'. An 'Export' button with a download icon is also present. The main content area is divided into two columns. The left column lists various technologies under several categories: Widgets (SoundCloud), Analytics (comScore, Facebook Pixel, Google Analytics, Quantcast Measure), Security (HSTS, DataDome, Quantcast Measure), and Miscellaneous (HTTP/2, Module Federation with a '50% sure' tag, Webpack, Open Graph, and PWA). The right column lists technologies under categories: CDN (Amazon CloudFront), JavaScript libraries (core-js 3.6.4), PaaS (Amazon Web Services), Cookie compliance (OneTrust), and Affiliate programs (Impact). Each technology is represented by a circular icon and a text label with a link.

Wappalyzer

TECHNOLOGIES MORE INFO Export

Widgets

- SoundCloud

Analytics

- comScore
- Facebook Pixel
- Google Analytics
- Quantcast Measure

Security

- HSTS
- DataDome
- Quantcast Measure

Miscellaneous

- HTTP/2
- Module Federation 50% sure
- Webpack
- Open Graph
- PWA

CDN

- Amazon CloudFront

JavaScript libraries

- core-js 3.6.4

PaaS

- Amazon Web Services

Cookie compliance

- OneTrust

Affiliate programs

- Impact

These are the technologies used.

Core.js is used and when checked against vulnerabilities, it shows that there is no direct vulnerability related to this version.

[Snyk Vulnerability Database](#) › [npm](#) › [core-js](#) › [core-js@3.6.4](#)

core-js@3.6.4 vulnerabilities

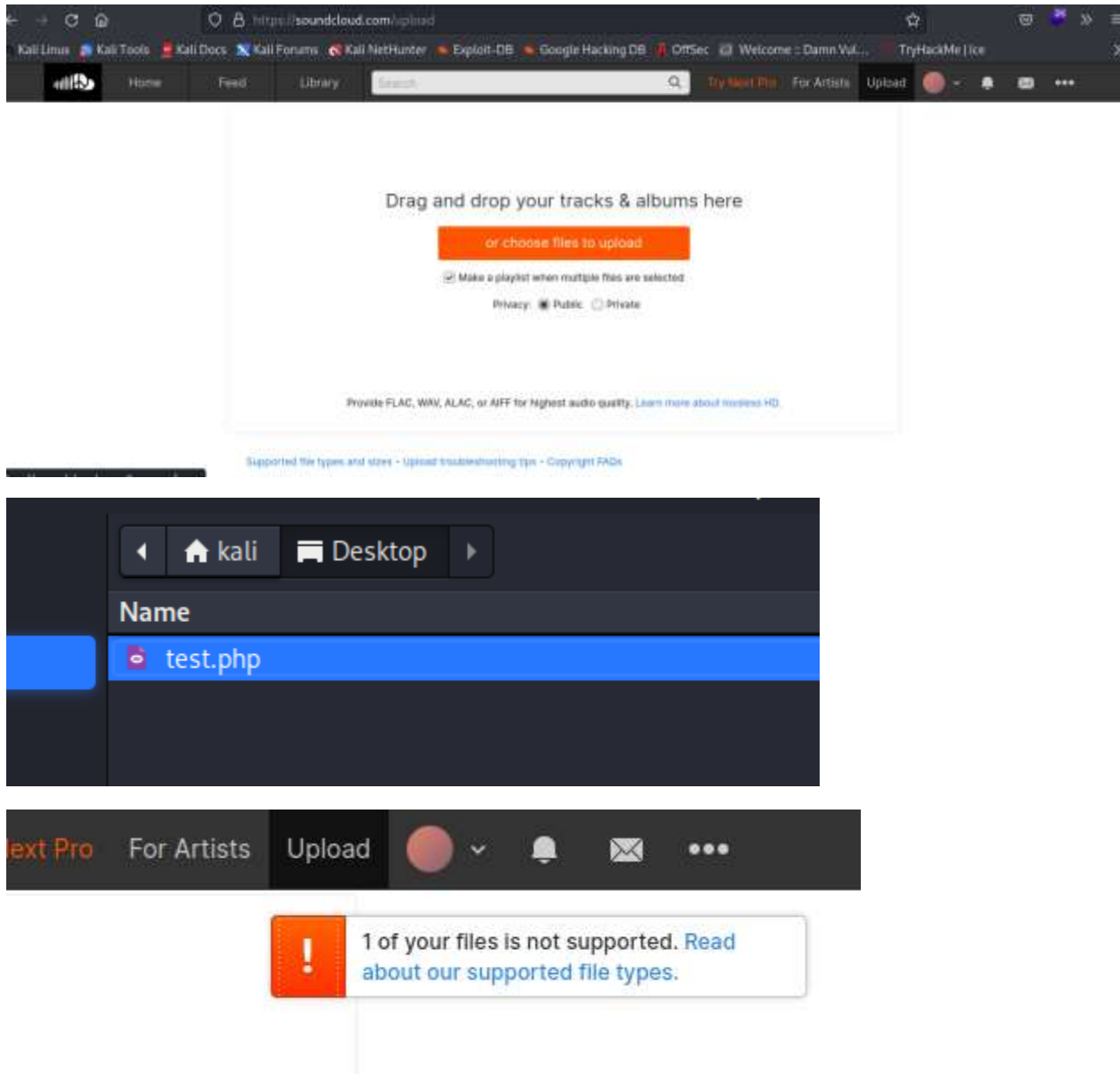
Standard library

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.

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.

Sqlmap

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

```
(kali㉿kali)-[~]
$ sqlmap -u https://soundcloud.com/search?q=hello12345

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It
state and federal laws. Developers assume no liability and are not responsible for any misuse or damage

[*] starting @ 12:57:28 /2024-05-02/

[12:57:30] [INFO] testing connection to the target URL
[12:57:31] [INFO] checking if the target is protected by some kind of WAF/IPS
[12:57:32] [INFO] testing if the target URL content is stable
[12:57:33] [WARNING] target URL content is not stable (i.e. content differs). sqlmap will base the page
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
[12:57:36] [INFO] testing if GET parameter 'q' is dynamic
[12:57:38] [WARNING] GET parameter 'q' does not appear to be dynamic
[12:57:38] [WARNING] heuristic (basic) test shows that GET parameter 'q' might not be injectable
[12:57:40] [INFO] testing for SQL injection on GET parameter 'q'
```

```
y
[12:59:28] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[12:59:37] [WARNING] GET parameter 'q' does not seem to be injectable
[12:59:37] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase
you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you
itch '--random-agent'
[12:59:37] [WARNING] your sqlmap version is outdated

[*] ending @ 12:59:37 /2024-05-02/
```

There is no injection vulnerability in the above web application.

Recon ng

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

```
[recon-ng][bb1][hackertarget] > options set SOURCE soundcloud.com
SOURCE ⇒ soundcloud.com
[recon-ng][bb1][hackertarget] > run

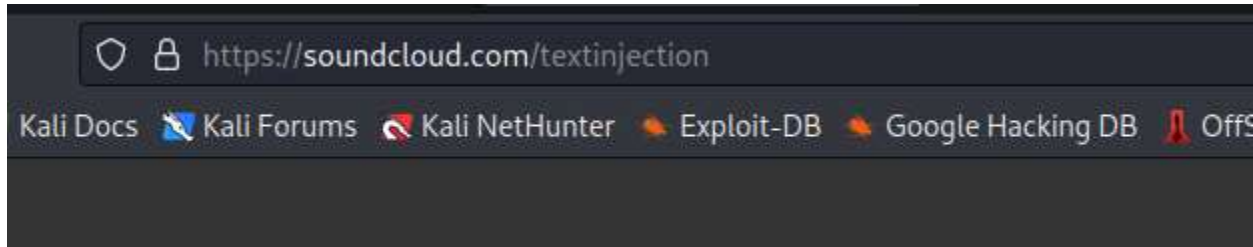
SOUNDCLOUD.COM
[*] Country: None
[*] Host: soundcloud.com
[*] Ip_Address: 18.154.185.39
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]
[*] Country: None
[*] Host: o13.account.soundcloud.com
[*] Ip_Address: 149.72.185.227
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]
[*] Ip_Address: 167.89.58.174
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]
[*] Country: None
[*] Host: o129.mail.announcements.soundcloud.com
[*] Ip_Address: 167.89.58.175
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]
[*] Country: None
[*] Host: o237.mail.announcements.soundcloud.com
[*] Ip_Address: 167.89.55.144
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]
[*] Country: None
[*] Host: o238.mail.announcements.soundcloud.com
[*] Ip_Address: 167.89.106.201
[*] Latitude: None
[*] Longitude: None
[*] Notes: None
[*] Region: None
[*]

SUMMARY
[*] 51 total (51 new) hosts found.
[recon-ng][bb1][hackertarget] >
```

51 total subdomains found.

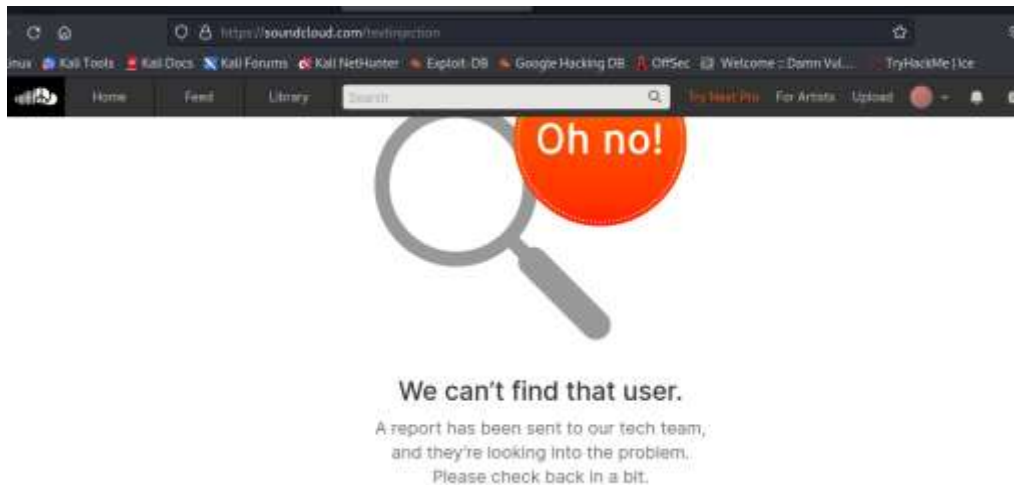
Text injection

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.

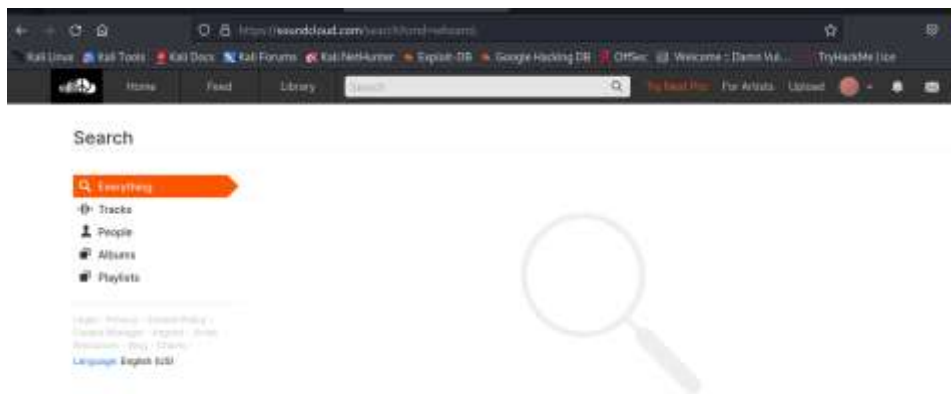
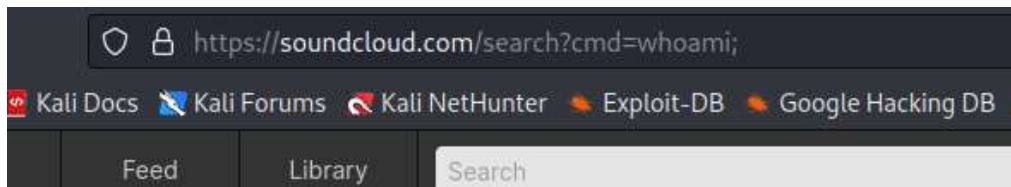
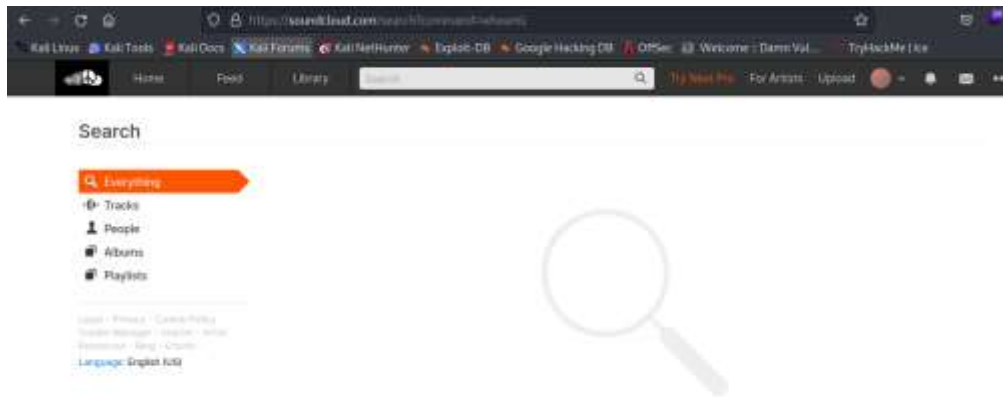
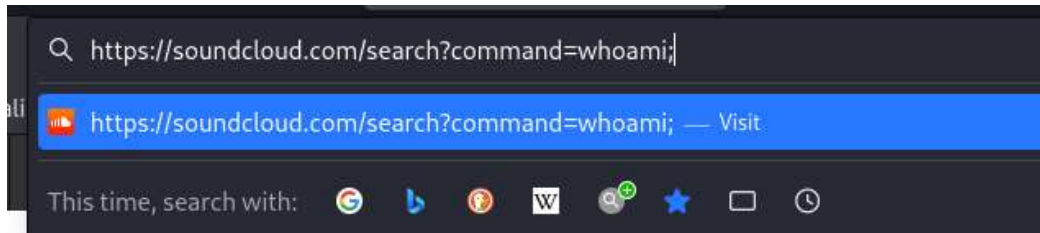


No text injection vulnerability can be found.

Command injection

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

The “whoami” command is appended to the url



No command injection vulnerability can be found.

Zap scan

Risk=High, Confidence=High (1)

<https://soundcloud.com> (1)

PII Disclosure (1)

- ▶ GET https://soundcloud.com/dj_bennett/vois-sur-ton-chemin-techno

Risk=Medium, Confidence=High (1)

<http://soundcloud.com> (1)

Content Security Policy (CSP) Header Not Set (1)

- ▶ GET <http://soundcloud.com>

Risk=Low, Confidence=High (2)

<https://soundcloud.com> (1)

Strict-Transport-Security Header Not Set (1)

- ▶ GET <https://soundcloud.com/george-ezra/green-green-grass>

<http://soundcloud.com> (1)

Server Leaks Version Information via "Server" HTTP Response Header Field (1)

- ▶ GET <http://soundcloud.com>

Risk=Low, Confidence=Medium (3)

<https://soundcloud.com> (1)

Application Error Disclosure (1)

- ▶ GET <https://soundcloud.com/xbox-app>

<http://soundcloud.com> (2)

Cross-Domain JavaScript Source File Inclusion (1)

- ▶ GET <http://soundcloud.com>

X-Content-Type-Options Header Missing (1)

- ▶ GET <http://soundcloud.com>

Risk=Low, Confidence=Low (1)

<http://soundcloud.com> (1)

Timestamp Disclosure - Unix (1)

- ▶ GET <http://soundcloud.com>

Risk=Informational, Confidence=Medium (2)

<http://soundcloud.com> (2)

Modern Web Application (1)

- ▶ GET <http://soundcloud.com>

User Agent Fuzzer (1)

- ▶ GET <http://soundcloud.com/alvinrisk>

Summary of zap scan:

- PII disclosure: PII data can be used to identify an individual therefore maintaining the data securely can mitigate risks.
- Content security policy (CSP) header not set: it works as an extra layer of security which should be set and configured correctly.
- Strict-transport-security header not set.
- Server leaks version information via “server” HTTP response header field- with the use of leaked data the attackers can exploit the vulnerable parts of the server.
- Application error disclosure: the warning messages disclose sensitive information which can be used to launch attacks by the attackers. A mechanism can be introduced which references the errors so it can solve this issue.
- Cross-domain javascript source file inclusion – it’s a warning. Happened when the external javascript is not validated.
- X-Content-Type-Option header is not set- allows to perform MIME-sniffing on the response body of old versions of chrome.
- Time stamp disclosure: