Bankacılıkta Admin Riskleri ve Bellekte Avcılık

Nebi Şenol YILMAZ
Danışman / Yönetici Ortak
senol.yilmaz@secrove.com

Secrove Information Security Consulting

Hakkımıda

- Nebi Şenol YILMAZ, CISA, CEH, 27001 LA
- Secrove Information Security Consulting
  - Bilgi Güvenliği Danışmanlığı
  - Bilgi Sistemleri Denetimi
  - Penetrasyon Testleri
  - Zafiyet Araştırması
  - Exploit Geliştirme
  - Bilgi Güvenliği Ar-Ge (Tubitak Destekli)

Ajanda

- Bankacılık
- Admin Riskleri
- Bellekte Avcılık

Bankacılık

Regülasyonlar

- PCI (W)
- SOX (US)
- CobIT (BDDK)
- İlkeler Tebliği (BDDK)
- İç Sistemler Yönetmeliği (BDDK)
- Destek Hizmetleri Yönetmeliği (BDDK)

...

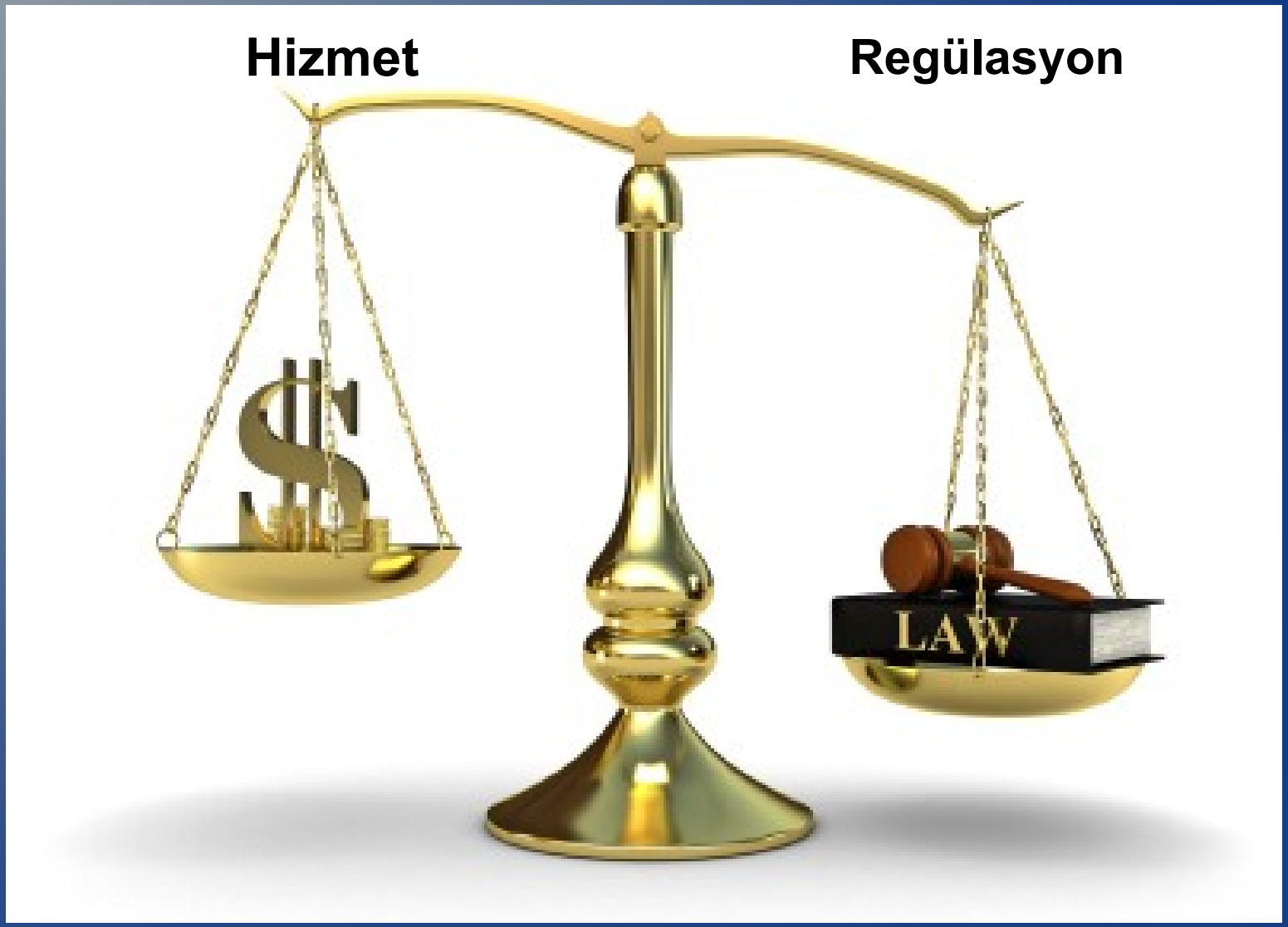
Bankacılık

- Teknoloji ile birlikte yeni projeler/ürünler oluşuyor.
- Yeni satış kanalları üretiliyor.

ve

- Her geçen gün regülasyonda değişiklik oluyor.
- Yeni riskler denetimleri gerektiriyor.

Bankacılık



Bankacılık

Özetle:

- Standartlar çerçevesinde önlem al!
- “Her işlemi” kayıt altında tut!

Bankacılık

Önlemler tamam da,

Neler kayıt altına alınacak...?

- Yetkilendirme logları
- Uygulama logları
- Veritabanı logları
- Trafik logları

...

Bankacılık

Unutulanlar:

- Admin'in çalıştırdığı komutlar
- Sniffer'lar
- Compiler'lar
- Debugger'lar
- Tracer'lar

...

Admin Riskleri

Bir sistem yöneticisi,

- Müşterilerin HTTP ile eriştikleri web sayfasındaki formlara girdikleri bilgileri nasıl görebilir?
  - Sniffer ?

Bellekte Avcılık

- Müşterilerin eriştiği web sayfalarını HTTPS yaptık... Yeterli mi?
- Ya da sistem yöneticisi sisteme SSH ile erişen kişilerin şifrelerini merak ediyorsa?

...

- Encrypted aktarılan bilgilerin riski, uygulama belleğinde devam ediyor...

Bellekte Avcılık

Process Trace (ptrace)

```
PTRACE(2)                  Linux Programmer's Manual                 PTRACE(2)

NAME
       ptrace - process trace

SYNOPSIS
       #include <sys/ptrace.h>

       long ptrace(enum __ptrace_request request, pid_t pid,
                   void *addr, void *data);

DESCRIPTION
       The ptrace() system call provides a means by which a parent process may
       observe and control the execution of another process, and  examine  and
       change its core image and registers.  It is primarily used to implement
       breakpoint debugging and system call tracing.
```

Bellekte Avcılık (ptrace)

- PTRACE_ATTACH
  - Process'e bağlanmak
- PTRACE_PEEKTEXT, PTRACE_PEEKDATA
  - Process belleğinden veri okumak
- PTRACE_POKETEXT, PTRACE_POKEDATA
  - Process belleğine veri yazmak
- PTRACE_DETACH
  - Process'ten ayrılmak

Bellekte Avcılık

Syscall Trace (strace)

```
STRACE(1)                                                    STRACE(1)

NAME
       strace - trace system calls and signals

SYNOPSIS
       strace  [  -dffhiqrtttTvxx  ] [ -acolumn ] [ -eexpr ] ...  [ -ofile ] [
       -ppid ] ...  [ -sstrsize ] [ -uusername ] [ -Evar=val ] ...  [ -Evar  ]
       ...  [ command [ arg ...  ] ]

       strace  -c  [ -eexpr ] ...  [ -Ooverhead ] [ -Ssortby ] [ command [ arg
       ...  ] ]

DESCRIPTION
       In the simplest case strace runs the specified command until it  exits.
       It  intercepts  and  records  the  system  calls  which are called by a
       process and the signals which are received by a process.  The  name  of
       each  system  call,  its  arguments and its return value are printed on
       standard error or to the file specified with the -o option.
```

Bellekte Avcılık (strace)

```
root@bt:~# cat catme
ABCDE
root@bt:~#
root@bt:~#
root@bt:~# strace cat catme
execve("/bin/cat", ["cat", "catme"], [/* 20 vars */]) = 0
brk(0)                                  = 0x80e8000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb77bb000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=69890, ...}) = 0
mmap2(NULL, 69890, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb77a9000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/tls/i686/cmov/libc.so.6", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0000m\1\0004\0\0\0"..., 512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1405508, ...}) = 0
mmap2(NULL, 1415592, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb764f000
mprotect(0xb77a2000, 4096, PROT_NONE)   = 0
mmap2(0xb77a3000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x153) = 0xb77a3000
mmap2(0xb77a6000, 10664, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb77a6000
close(3)                                = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb764e000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb764e8d0, limit:1048575, seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0
mprotect(0xb77a3000, 8192, PROT_READ)   = 0
mprotect(0x8054000, 4096, PROT_READ)    = 0
mprotect(0xb77d9000, 4096, PROT_READ)   = 0
munmap(0xb77a9000, 69890)               = 0
brk(0)                                  = 0x80e8000
brk(0x8109000)                          = 0x8109000
fstat64(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 0), ...}) = 0
open("catme", O_RDONLY|O_LARGEFILE)     = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=6, ...}) = 0
read(3, "ABCDE\n", 32768)               = 6
write(1, "ABCDE\n", 6ABCDE
)                     = 6
read(3, "", 32768)                      = 0
close(3)                                = 0
close(1)                                = 0
close(2)                                = 0
exit_group(0)                           = ?
root@bt:~#
```

Bellekte Avcılık (strace)

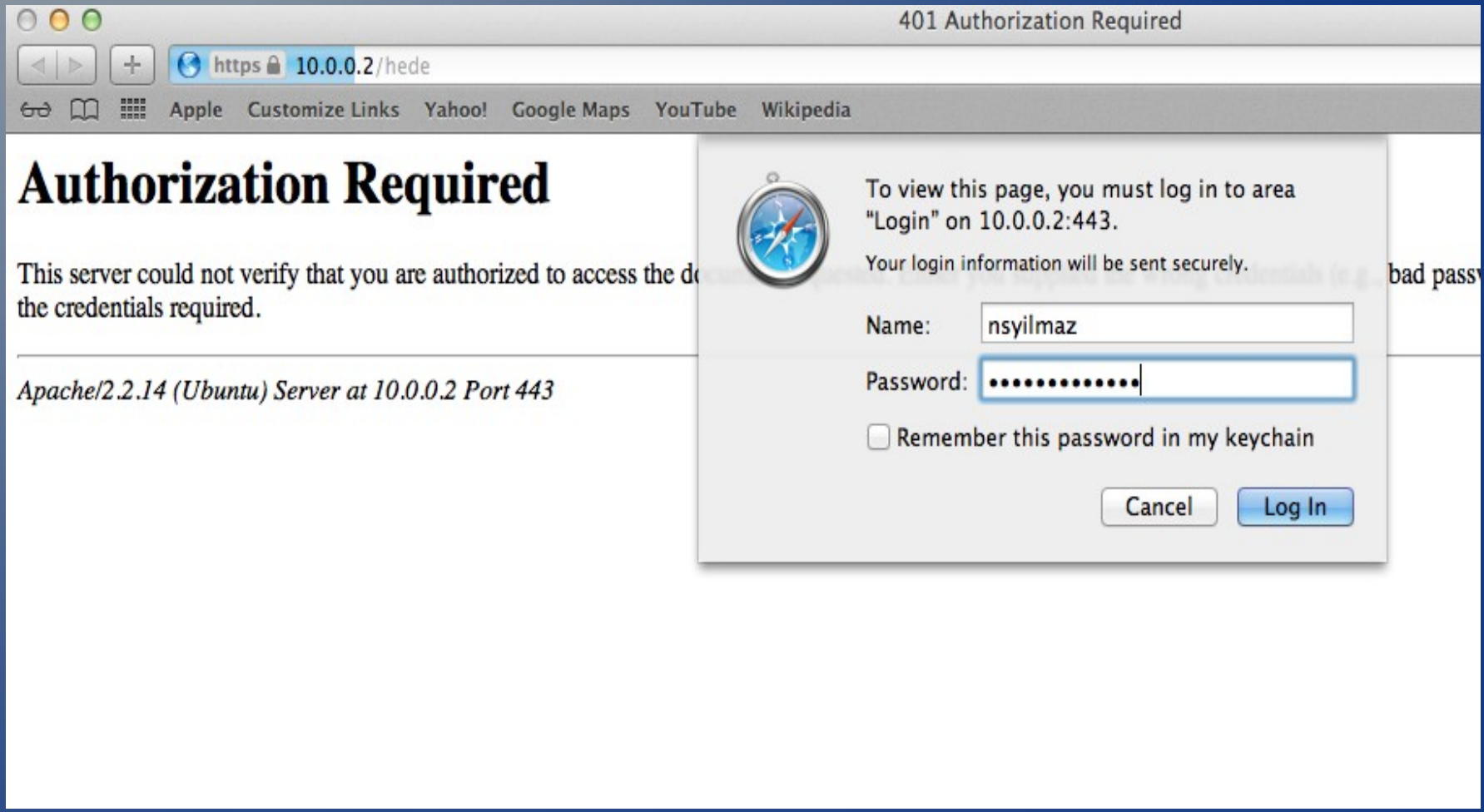
“cat” komutunu trace ettik...

Peki çalışmakta olan SSL ve SSH process'lerini nasıl trace edeceğiz?

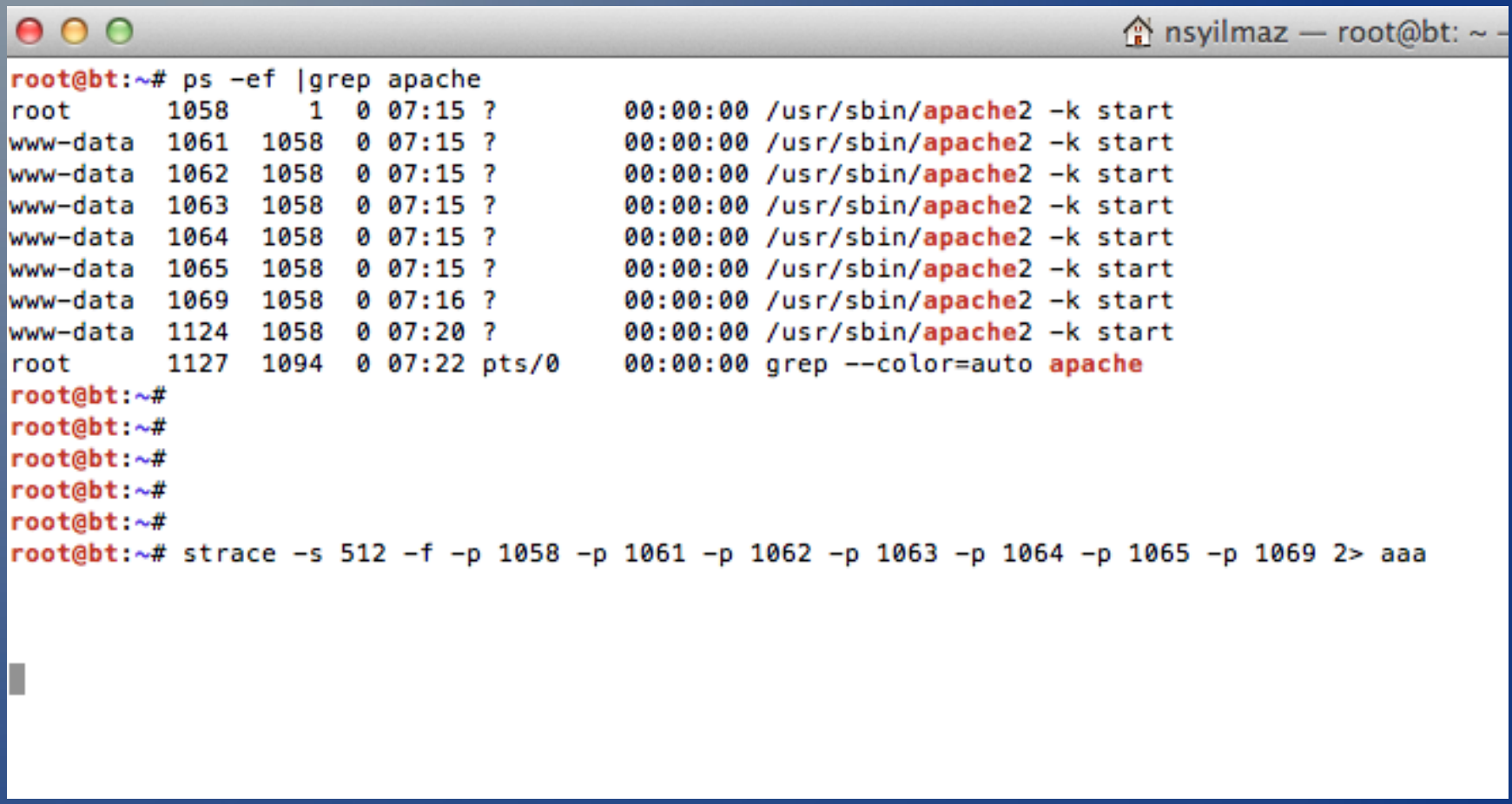
Bellekte Avcılık (strace)
- Uygulama 1 -

SSL olarak çalışan apache processlerini trace edelim:

Bellekte Avcılık (strace)
- Uygulama 1 -



Bellekte Avcılık (strace)
- Uygulama 1 -



Bellekte Avcılık (strace)
- Uygulama 1 -

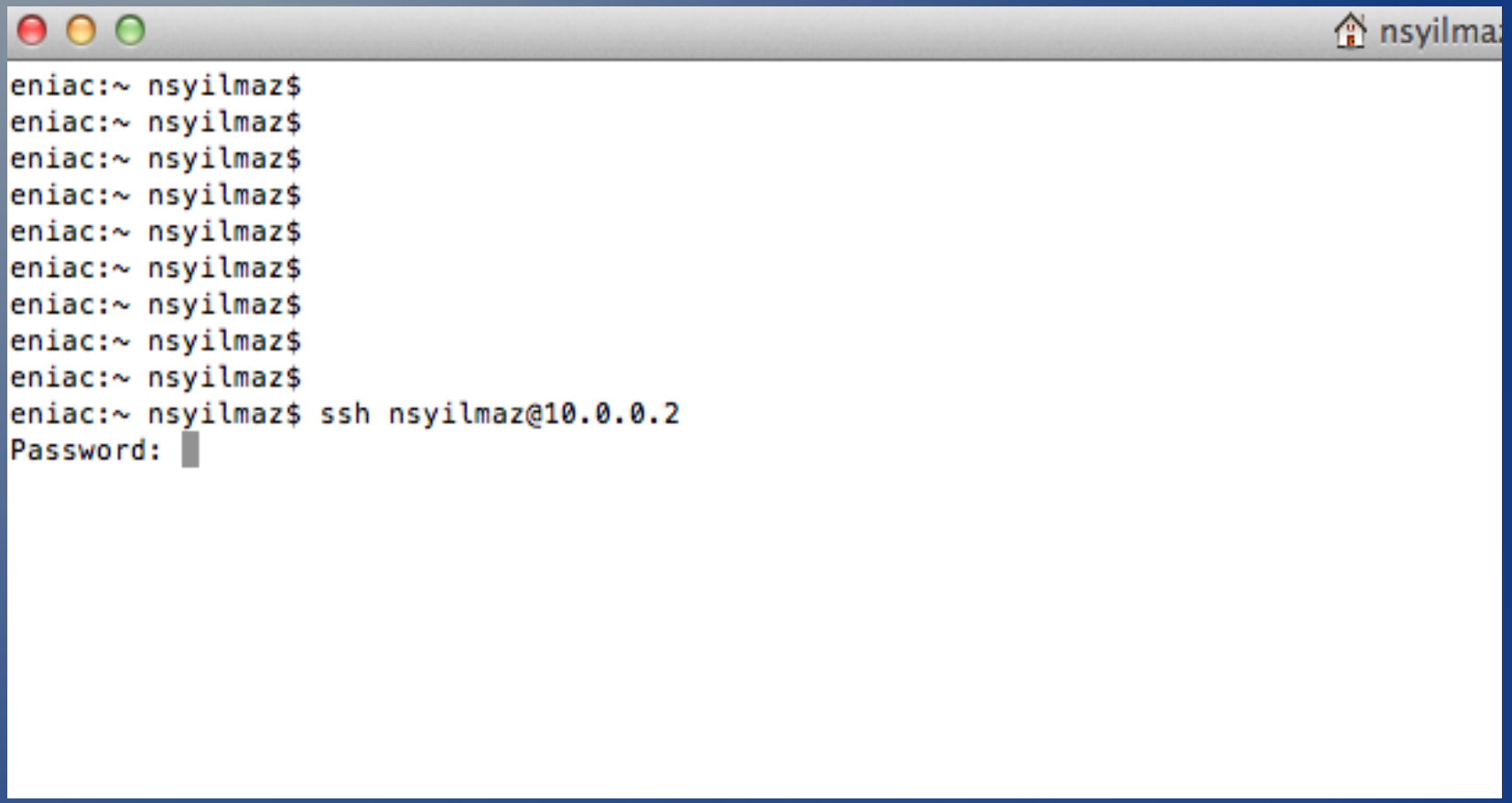
nsyilmaz — root@bt: ~ — ssh — 181×49

```
[pid  1064] gettimeofday({1352006551, 547771}, NULL) = 0
[pid  1064] gettimeofday({1352006551, 547837}, NULL) = 0
[pid  1064] gettimeofday({1352006551, 547921}, NULL) = 0
[pid  1064] stat64("/var/www/hede", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
[pid  1064] open("/var/www/hede/.htaccess", O_RDONLY|O_LARGEFILE) = 13
[pid  1064] fcntl64(13, F_GETFD)        = 0
[pid  1064] fcntl64(13, F_SETFD, FD_CLOEXEC) = 0
[pid  1064] fstat64(13, {st_mode=S_IFREG|0644, st_size=117, ...}) = 0
[pid  1064] read(13, "AuthType Basic\nAuthName \"Login\"\nAuthBasicProvider external\nAuthExternal pwauth\nAuthzUnixgroup
[pid  1064] read(13, "", 4096)          = 0
[pid  1064] close(13)                   = 0
[pid  1064] pipe([13, 14])              = 0
[pid  1064] fcntl64(14, F_GETFD)        = 0
[pid  1064] fcntl64(14, F_SETFD, FD_CLOEXEC) = 0
[pid  1064] rt_sigaction(SIGCHLD, {SIG_DFL, [], SA_INTERRUPT}, {SIG_DFL, [], 0}, 8) = 0
[pid  1064] clone(Process 1130 attached (waiting for parent)
Process 1130 resumed (parent 1064 ready)
child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD, child_tidptr=0xb7451768) = 1130
[pid  1064] close(13 <unfinished ...>
[pid  1130] close(10 <unfinished ...>
[pid  1064] <... close resumed> )       = 0
[pid  1064] write(14, "nsyilmaz", 8)    = 8
[pid  1064] write(14, "\n", 1)          = 1
[pid  1064] write(14, "CokGizliSifre", 13) = 13
[pid  1064] write(14, "\n", 1)          = 1
[pid  1064] close(14)                   = 0
[pid  1064] waitpid(1130, Process 1064 suspended
 <unfinished ...>
[pid  1130] <... close resumed> )       = 0
[pid  1130] close(4)                    = 0
[pid  1130] close(3)                    = 0
[pid  1130] close(9)                    = 0
```

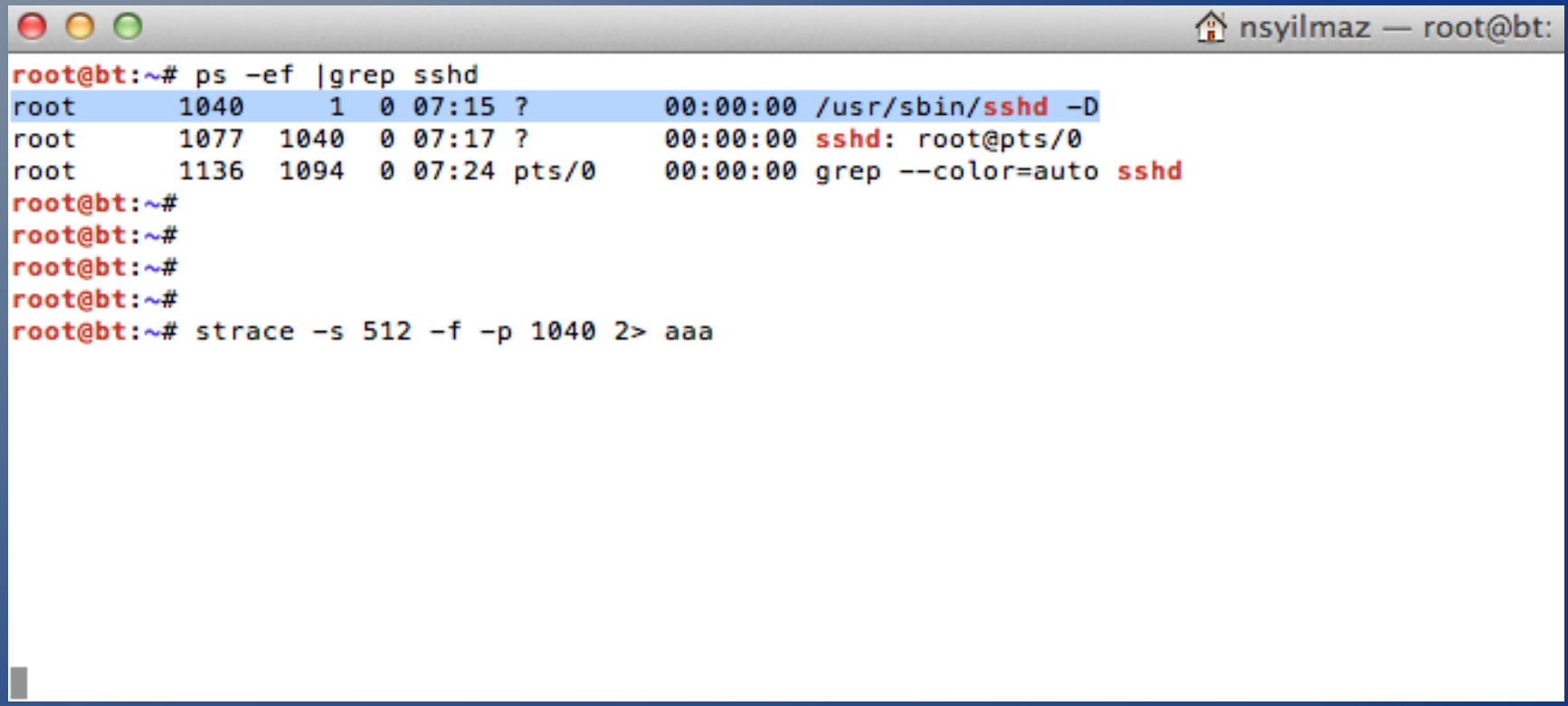
Bellekte Avcılık (strace)
- Uygulama 2 -

SSH processini trace edelim:

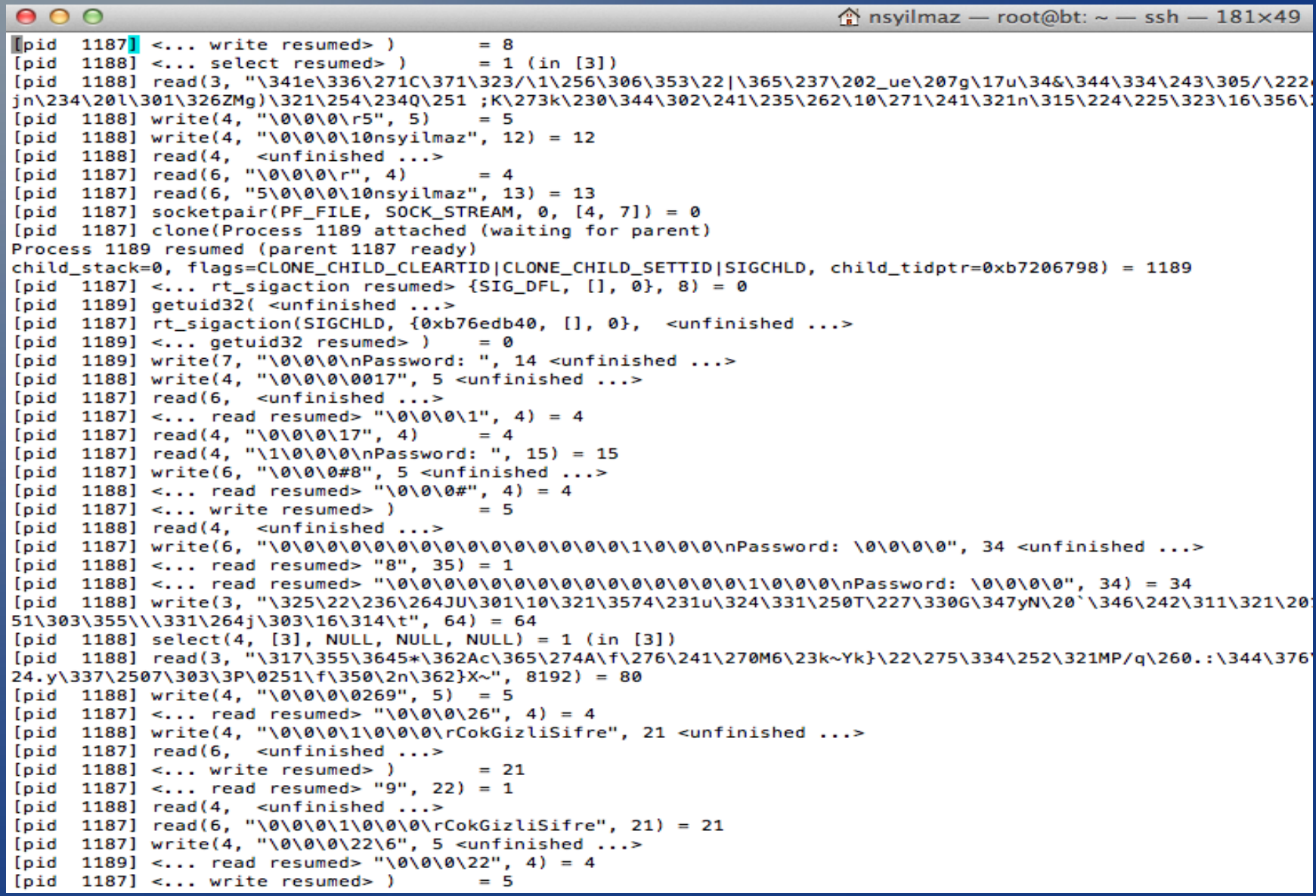
Bellekte Avcılık (strace)
- Uygulama 2 -



Bellekte Avcılık (strace)
- Uygulama 2 -



Bellekte Avcılık (strace)
- Uygulama 2 -



Özet

- Bellekte olan bellekte kalsın :)
- Standart olarak uygulamaların logları tutulmalı

  ancak,
- Bu logların yetersiz olduğu bilinmeli ve daha alt seviyede her işlem (komut vs.) mutlaka kayıt altına alınmalı ve gözden geçirilmelidir.

Teşekkürler !

senol.yilmaz@secrove.com

@nsyilmaz

nsyilmaz