Roubando sudo con ptrace

Eloy Pérez González
@zer1t0@defcon.social

Hackliza!

sudo

- Permite executar comandos como “root“ (ou outro usuario)
- Pode pedir o contrasinal do usuario (non a de root)
- Mantén unha caché de autenticacións (15 minutos por defecto)

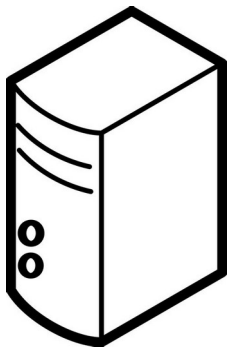
```
$ whoami
it
$ sudo whoami
[sudo] password for it:
root
```

Situación

```
$ ssh -i itkey it@itserver
it@itserver:~$ sudo id
[sudo] password for it:
```

????

hackerman



```
$ ssh it@itserver
it@itserver:~$ sudo whoami
[sudo] password for it:
root
```

it admin

Demo time

- Sesións de sudo (de sudoers realmente):
  + 15 minutos (por defecto)
  + Gárdanse en */run/sudo/ts* → ficheiro por usuario

```
struct timestamp_entry {
        unsigned short version;     /* version number */
        unsigned short size;        /* entry size */
        unsigned short type;        /* TS_GLOBAL, TS_TTY, TS_PPID */
        unsigned short flags;       /* TS_DISABLED, TS_ANYUID */
        uid_t auth_uid;             /* uid to authenticate as */
        pid_t sid;                  /* session ID associated with tty/ppid */
        struct timespec start_time; /* session/ppid start time */
        struct timespec ts;         /* time stamp (CLOCK_MONOTONIC) */
        union {
            dev_t ttydev;           /* tty device number */
            pid_t ppid;             /* parent pid */
        } u;
    };
```

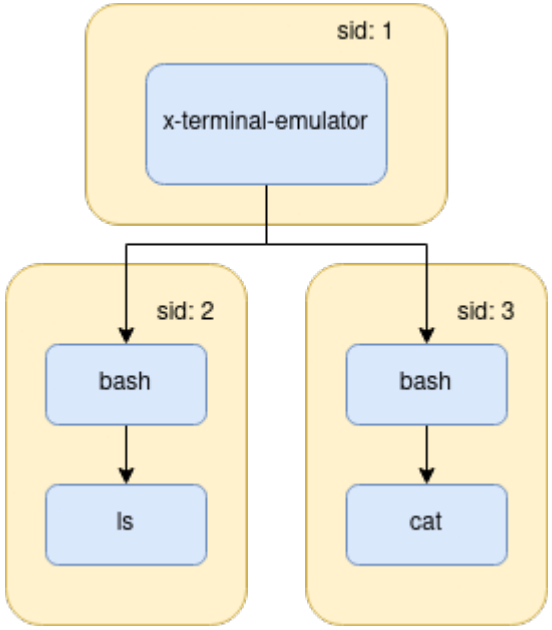
```
$ sudo ./sudohunt read
......
version: 2
size: 56
type: 2 TS_TTY
flags: 0
auth_uid: 1000 user
sid: 4867 /bin/bash
start_time: 2543.490000000 (754.691432538 seconds ago)
ts: 2563.851568651 (734.329863887 seconds ago)
tty: 34816 /dev/pts/0
```

Sesión de procesos

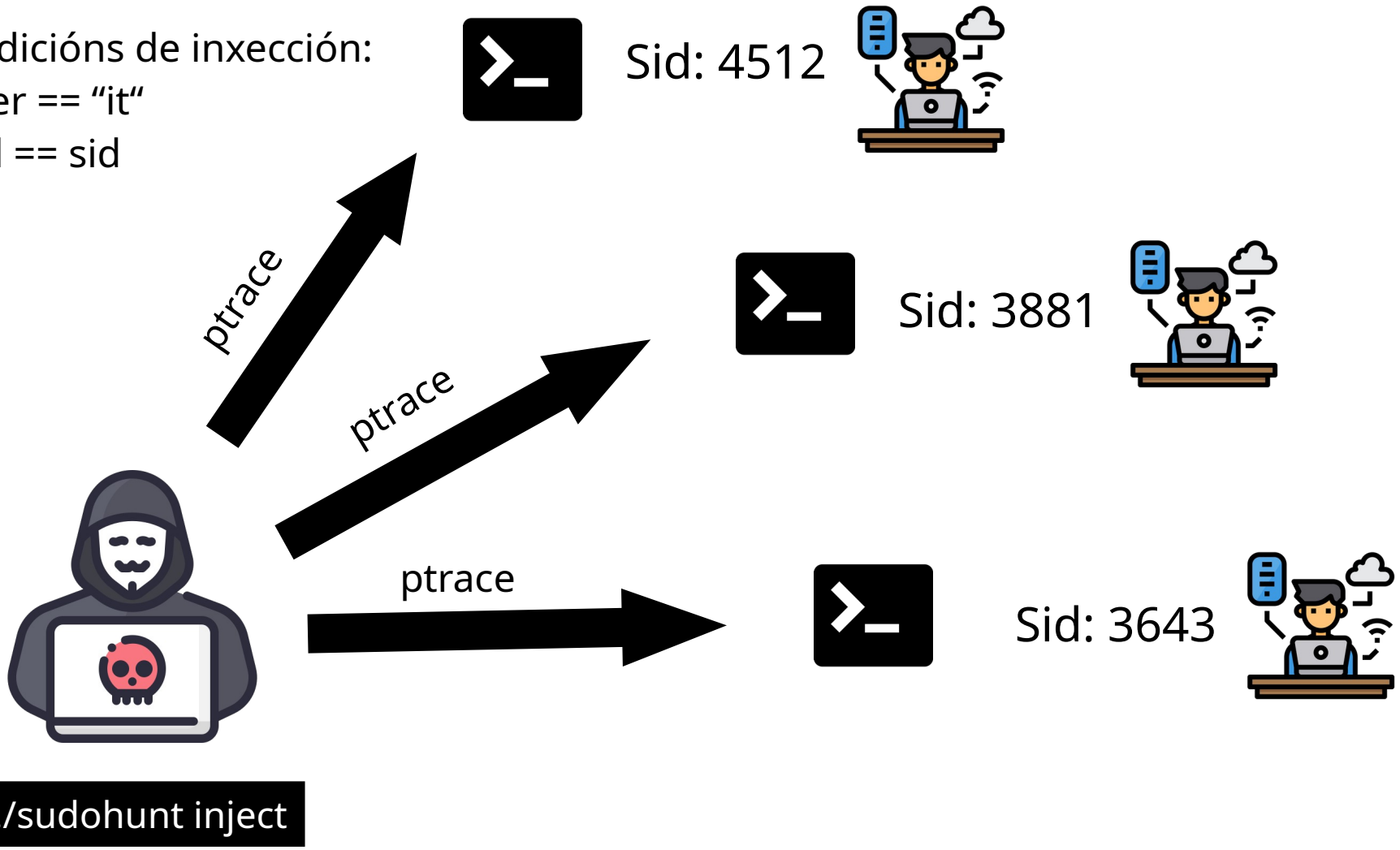
- Agrupación de procesos emparentados

- Todo proceso está na sesión do pai se non cambia (invoca a setsid)

- Sesión de procesos != sesión de usuario



Condicións de inxección:
- user == “it“
- pid == sid




```
$ ./sudohunt inject
```

ptrace

- Syscall para acoplarse (trace) a procesos

- Permite:
  + Leer memoria
  + Controlar execución
  + Modificar rexistros
  + etc, etc, etc

- Usado por debuggers (gdb), tracers (strace), etc

- Controlase o alcance con `/proc/sys/kernel/yama/ptrace_scope`

Sid: 1337

$ ./sudohunt inject

1. ptrace(attach)

Sid: 4512

Sid: 1337

$ ./sudohunt inject

3. ptrace(detach)

Sid: 4512

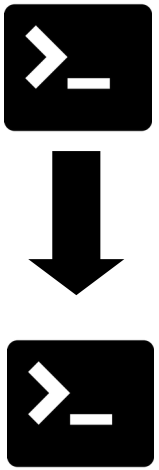
2. fork

Sid: 1337

```
$ ./sudohunt inject
```

4. ptrace(attach)

Sid: 4512



Sid: 1337

```
$ ./sudohunt inject
```

5. ptrace(detach)

Sid: 4512

6.execve

```
sudo -n sudohunt write –pid 1337
```

Sid: 1337

Pwned!!

```
$ sudo whoami
root
```

Sid: 4512

Como nos protexemos?

`/proc/sys/kernel/yama/ptrace_scope` != 0

- 0: Sen restriccións
- 1: So procesos fillos (Eg: gdb) ou CAP_SYS_PTRACE
- 2: Precisas CAP_SYS_PTRACE
- 3: Non funciona ptrace

Por defecto:
- Ubuntu : 1
- Debian 12: 0
- Alma Linux 9: 0

Referencias

- sudohunt : https://gitlab.com/Zer1t0/sudohunt

- sudo inject: https://github.com/nongiach/sudo_inject

- Yama ptrace scope:
https://www.kernel.org/doc/html/latest/admin-guide/LSM/Yama.html

- sudoers timestamp:
https://www.sudo.ws/docs/man/sudoers_timestamp.man/

- ptrace: https://www.man7.org/linux/man-pages/man2/ptrace.2.html

Graciñas...

...e bo hacking!!