Security

Suid e capabilites

- find / -perm +6000 -type f -user root
  - In generale il controllo di root è assoluto. Ha senso lasciar girare un webserver da root solo perché deve utilizzare una porta privilegiata?
  - **`man capability`**
    - CAP_NET RAW : allows to use raw sockets
    - CAP_NET ADMIN : allows to change routing tables
    - CAP_KILL : unlocks signal sending to everyone
    - CAP_SYS NICE : allows to renice with negative values
  - **`chmod u-s /bin/ping`**
  - **`setcap cap_net_raw+ep /bin/ping`**

Suid e capabilites

Capability Sets

Each thread has three capability sets containing zero or more of the above capabilities:

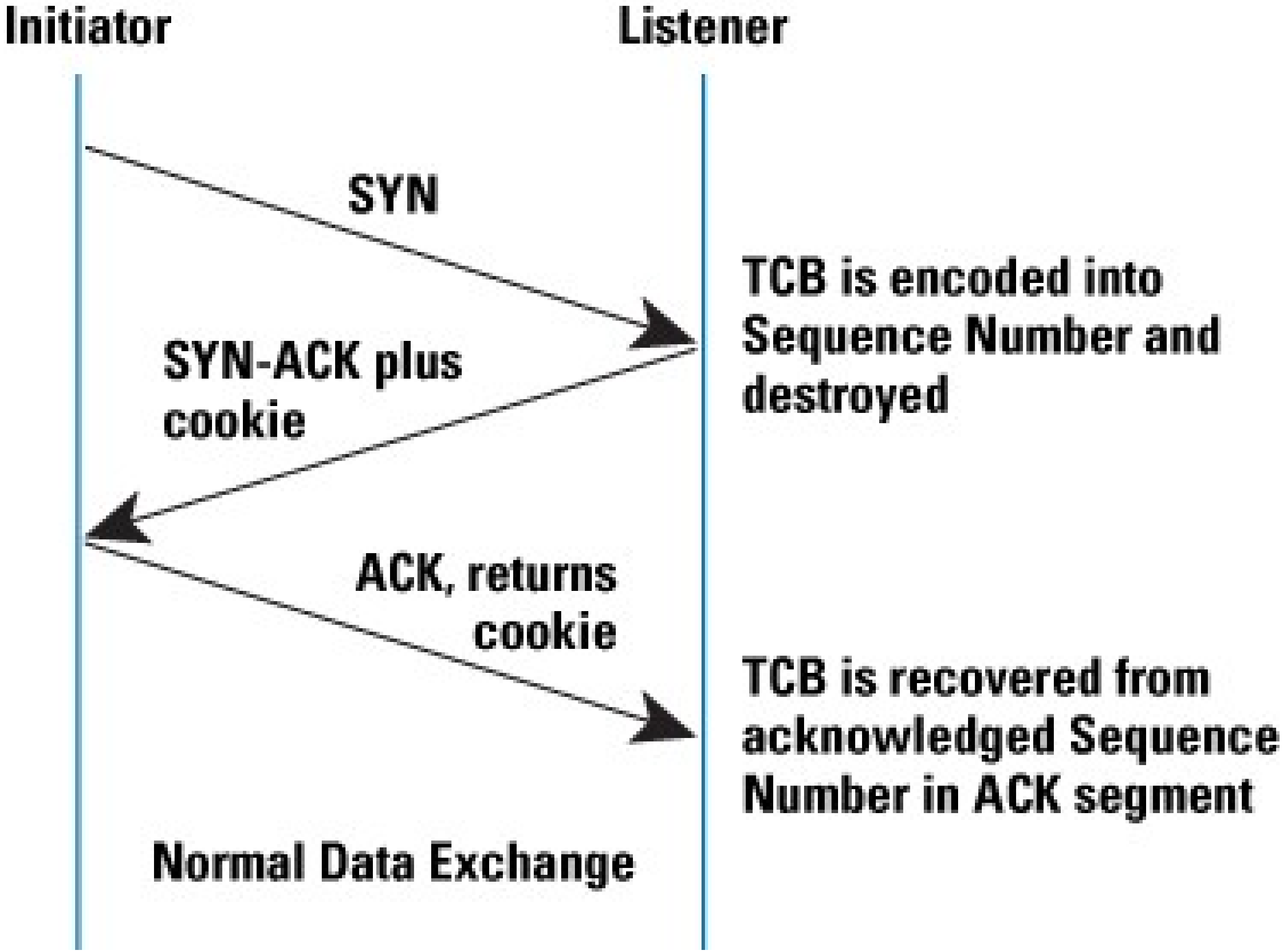
Effective - the capabilities used by the kernel to perform permission checks for the thread.

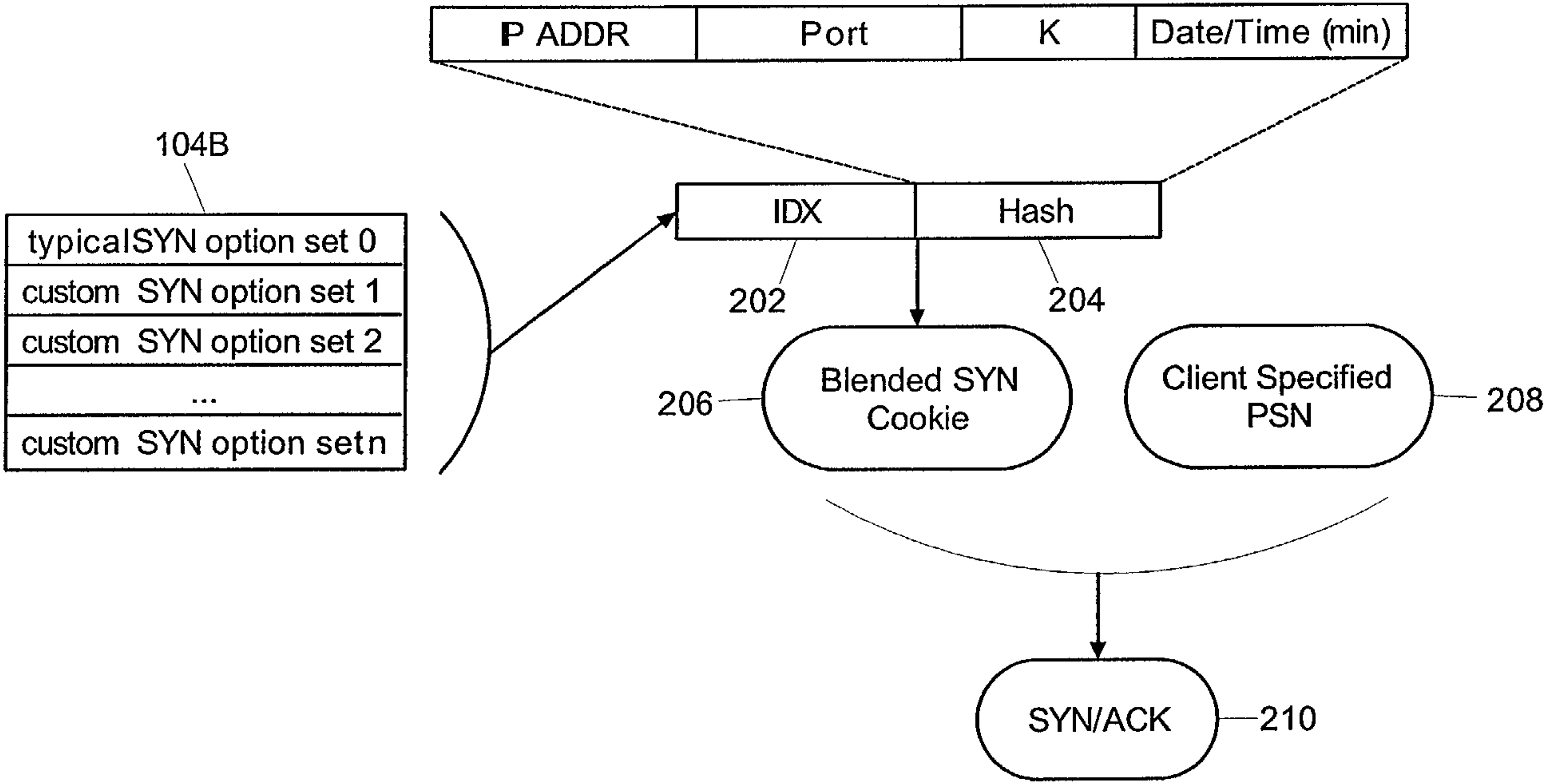
Permitted - the capabilities that the thread may assume (i.e., a limiting superset for the effective and inheritable sets). If a thread drops a capability from its permitted set, it can never re-acquire that capability (unless it exec()s a set-user-ID-root program).

inheritable - the capabilities preserved across an execve(2). A child created via fork(2) inherits copies of its parent's capability sets. See below for a discussion of the treatment of capabilities during exec(). Using capset(2), a thread may manipulate its own capability sets, or, if it has the CAP_SETPCAP capability, those of a thread in another process.

SYNCookies

- Sono utilizzati per evitare i ddos
  - Come funziona un ddos? Perché un ddos ci dà problemi?
  - Trasformare l'occupazione in memoria in una occupazione di cpu
  - **`echo 1 > /proc/sys/net/ipv4/tcp_syncookies`**
  -
- In generale, per ottimizzazioni e sicurezza di rete, conviene controllare i vari parametri presenti in: **`/proc/sys/net/`**


Initiator
Listener
SYN
TCB is encoded into Sequence Number and destroyed
SYN-ACK plus cookie
ACK, returns cookie
TCB is recovered from acknowledged Sequence Number in ACK segment
Normal Data Exchange


IP ADDR
Port
K
Date/Time (min)
104B
typical SYN option set 0
custom SYN option set 1
custom SYN option set 2
...
custom SYN option set n
IDX
Hash
202
204
206
Blended SYN Cookie
Client Specified PSN
208
SYN/ACK
210

Qualche consiglio sui database

- Utilizzare un utente diverso per ogni servizio
- Controllare i permessi assegnati a quell'utente di modo che non possa fare danni
- Non lasciare accesso al database via remoto (bloccare sul firewall e nella config del database)
- Vediamo un esempio

fail2ban

```
    [DEFAULT]

    # “ignoreip” can be an IP address, a CIDR mask or a DNS host
    ignoreip = 127.0.0.1 172.31.0.0/24 10.10.0.0/24
192.168.0.0/24
    bantime = 86400
    maxretry = 5

    [ssh-iptables]
    enabled = true
    filter = sshd
    action = iptables[name=SSH, port=ssh, protocol=tcp]
    sendmail-whois[name=SSH, dest=you@mail.com,
sender=fail2ban@mail.com]
    logpath = /var/log/auth.log
    maxretry = 5

    logpath=/var/log/secure (for RedHat,CentOS,Fedora)
```

fail2ban

- Si può utilizzare anche per altri servizi oltre a ssh, ad esempio un server SMTP/IMAP. Bisogna smanettare un po' sulla configurazione.

Chroot

- Comodo per far girare i programmi come se fossero in una installazione diversa del sistema
- I programmi possono sempre accedere a processi, device,...
- Più che altro separiamo la partizione di root (/) da quella principale e usare un utente diverso da root.
- Gira tutto sotto lo stesso kernel
- chroot /cartella /bin/bash

Un paio di tuning su ssh

- Forzare il protocollo versione 2:
  - “Protocol 2”
- Disaiblitare login per root:
  - “PermitRootLogin no”
- Forzare l'auth con chiave pubblica:
  - “PubkeyAuthentication yes”
  - “PasswordAuthentication no”
  - Se usate solo chiavi pubbliche può aver senso togliere le password per gli utenti e usare solo “sudo”

Come fare un chroot dentro ssh, disabilitare forwarding

```
#sftp per far funzionare chroot
Subsystem     sftp   internal-sftp

Match User utente
        ChrootDirectory /home/utente/sec/
        ForceCommand internal-sftp
        X11Forwarding no
        AllowTcpForwarding no
        PermitOpen 127.0.0.1:3306
```

Nota: la cartella in cui facciamo chroot deve avere root come proprietario

A livello di gruppo

```
#sftp per far funzionare chroot
Subsystem     sftp   internal-sftp

Match Group ssh-chroot
    ChrootDirectory %h
    ForceCommand internal-sftp
    AllowTcpForwarding no
    X11Forwarding no
```

Nota: la cartella in cui facciamo chroot deve avere root come proprietario

Usare i namespace (LXC)

- **`unshare -n bash`**
  - Lancia un processo bash in un namespace di rete separato
  - `...`
- È possibile costruirsi delle “macchine virtuali” complete utilizzando LXC che si basa sui namespace di rete
- Per macchine virtuali complete: QEMU

Protezione dello stack

- /proc/sys/kernel/randomize_va_space
-

```
PaX Control ->

 [ ] Support soft mode
 [ ] Use legacy ELF header marking
 [ ] Use ELF program header marking
 [*] Use filesystem extended attributes marking
     MAC system integration (none)  --->
```

```
File systems  --->
 <*> The Extended 4 (ext4) filesystem
 -*-   Ext4 extended attributes
 [ ]     Ext4 POSIX Access Control Lists
 [ ]     Ext4 Security Labels
 [ ]   EXT4 debugging support
```

```
PaX control v0.7
Copyright 2004,2005,2006,2007,2......

usage: paxctl <options> <files>

options:
   -p: disable PAGEEXEC      -P: enable PAGEEXEC
   -e: disable EMUTRAMP      -E: enable EMUTRAMP
   -m: disable MPROTECT      -M: enable MPROTECT
   -r: disable RANDMMAP      -R: enable RANDMMAP
   -x: disable RANDEXEC      -X: enable RANDEXEC
   -s: disable SEGMEXEC      -S: enable SEGMEXEC

   -v: view flags       -z: restore default flags
   -q: suppress error messages-Q: report flags in
short format
   -c: convert PT_GNU_STACK into PT_PAX_FLAGS (see
manpage!)
   -C: create PT_PAX_FLAGS (see manpage!)
```

- PAGEEXEC:
  - The kernel will protect non-executable pages based on the paging feature of the CPU. This is sometimes called "marking pages with the NX bit" in other OSes. This feature can be controlled on a per ELF object basis by the PaX P and p flags.
- EMUTRAMP:
  - The kernel will emulate trampolines (snippets of executable code written on the fly) for processes that need them, eg. nested functions in C and some JIT compilers. Since trampolines try to execute code written by the process itself to memory marked as non-executable by PAGEEXEC or SEGMEXEC, the PaX kernel would kill any process that tries to make use of one. EMUTRAMP allows these processes to run without having to fully disable enforcement of non-executable memory. This feature can be controlled on a per ELF object basis by PaX E and e flag.

- MPROTECT:
  - The kernel will prevent the introduction of new executable pages into the running process by various techniques: it will forbid the changing of the executable status of pages, or the creation of anonymous RWX mappings, or making RELRO data pages as writable again. It is controlled on a per ELF object basis by the PaX M and m flag.
- RANDMMAP:
  - The kernel will use a randomized base address for mmap() requests that do not specify one via the MAP_FIXED flag. It is controlled by the PaX R and r flags.

- RANDEXEC:
  - The goal of RANDEXEC is to introduce randomness into the main executable file mapping addresses.
- SEGMEXEC:
  - This is like PAGEEXEC, but based on the segmentation feature of the CPU and it is controlled by the PaX S and s flags. Note that SEGMEXEC is only available on CPUs that support memory segmentation, namely x86.

```
# paxctl -v /usr/bin/python3.2
PaX control v0.7
Copyright 2004,2005,2006,2007,2009,2010,2011,2012 PaX Team
<pageexec@freemail.hu>

- PaX flags: -----m-x-e-- [/usr/bin/python3.2]
   MPROTECT is disabled
   RANDEXEC is disabled
   EMUTRAMP is disabled

# paxctl -P /usr/bin/python3.2
# paxctl -v /usr/bin/python3.2
PaX control v0.7
Copyright 2004,2005,2006,2007,2009,2010,2011,2012 PaX Team
<pageexec@freemail.hu>

- PaX flags: P----m-x-e-- [/usr/bin/python3.2]
   PAGEEXEC is enabled          <--- Note: this added to
the earlier flags, it didn't overwrite them.
   MPROTECT is disabled
   RANDEXEC is disabled
   EMUTRAMP is disabled
```

```
# getfattr -n user.pax.flags /usr/bin/python3.2
getfattr: Removing leading '/' from absolute path
names
# file: usr/bin/python3.2
user.pax.flags="em"

# setfattr -n user.pax.flags -v P /usr/bin/python3.2
# getfattr -n user.pax.flags /usr/bin/python3.2
getfattr: Removing leading '/' from absolute path
names
# file: usr/bin/python3.2
user.pax.flags="P"
```

Domande?

Grazie!

Mail: daniele punto iamartino chiocciola mail punto polimi punto it