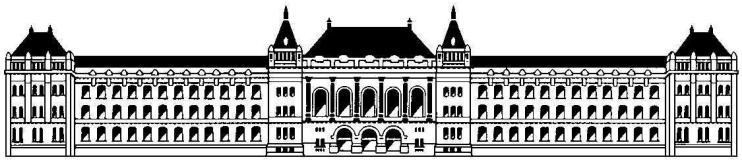


Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek tanszék

IT Infrastukturális elemek függőségeinek felderítése

Önálló laboratórium
2010/2

Önálló Laboratórium

Készítette:

Orova Péter (E6P9SA)

Konzulens:

Kocsis Imre

1 Bevezetés

1.1 A kutatás célja

Egy összetett IT rendszerben óhatatlanul előfordul az, hogy egyes alkotóelemei bizonyos oknál fogva nem az elvárt működés szerint viselkednek, hibás kimenetet produkálnak illetve előre nem látható módon leállnak. Ezen hibák az IT rendszer többi elemére is hatással lehetnek, ám ezen hatások előrejelzése a jelenlegi lehetőségek mellett nem megoldott. Ahhoz, hogy az ilyen és hasonló jellegű egymásrahatásokat előre tudjunk jelezni, az elemek közti függőségeket kell feltérképeznünk.

A függőségeket két nagyobb kategóriába sorolhatjuk, egyrészről adatfolyam, másrészről erőforrás jellegűekről beszélhetünk. Erőforrás jellegű függésről akkor beszélünk amikor egy elem működése szükséges a függő elem helyes működéséhez, például egy webalkalmazás erőforrás jellegű függésben áll a hálózati kapcsolattal. Adatfolyam jellegű függőségről akkor beszélünk amikor egy elem által produkált kimenetet egy másik elem felhasználja. Tipikus esete ez utóbbinak, mikor egy program egy fájlba ír, majd egy másik onnan kiolvassa az adatot.

1.2 Probléma

A fenti bevezető alapján látható, hogy igen finom granularitású függőségeket kell számontartani illetve felderíteni egy rendszeren belül. A tipikusan hasonló célra (is) jelenleg alkalmazott CMDB technológia ezt nem teszi lehetővé. Célunk tehát, hogy egy olyan megoldást találjunk mely segítségével megfelelő granularitással megfigylehetőek a fentebb említett függőségek.

1.3 Alapötlet

Egy IT rendszerben éles, business környezetben, jellemzően biztosnági megfontolásokból fut valamiféle MAC/DAC rendszer is, mellyel az egyes programok jogosultságait kezelik, illetve kényszerítik ki. Általánosságban az ilyen rendszerek a megfigyelt szoftver minden egyes erőforrás- és fájlhozzáférési kísérletét képesek érzékelni majd a szoftver jogosultsága szerint azt engedélyezni vagy letiltani.

Ezen MAC/DAC rendszerek közös tulajdonsága továbbá, hogy létezik olyan felderítő működési módjuk, amikor a megfigyelt programok hozzáféréseit nem korlátozzák, de minden esetben jegyzik. Ez az a tulajdonságuk, mely a kutatás számára rendkívül ígéretessé teszi őket, ugyanis ezt kihasználva lehetőségünk nyílna a fent említett nagyon finom granularitású függéseket is megfigyelni.

A félév során vizsgáltunk továbbá alkalmazásfüggetlen diagnosztikai eszközöket is.

2 Vizsgált eszközök

2.1 Lsof

2.1.1 Eszköz leírása

Az lsof parancs segítségével a linux rendszereken jeleníthetjük meg a megnyitott fileokat, és az őket megnyitó processzeket. A kutatás témája szempontjából azért érdemes ezzel az eszközzel foglalkozni mert segítségével az egyes processzek, ezen keresztül az őket indító végrehajtható binárisok és fájlok között vagyunk képesek függőséget megállapítani.

Az ezen eszköz által felderíthető fájltípusok linux rendszeren: közönséges fájlok, IP Socketek, valamint pipe-ok.

2.1.2 Használat

Az önálló laboratórium során ezt a programot a következőképpen használtuk:

```
lsof
```

kimenet (részlet):

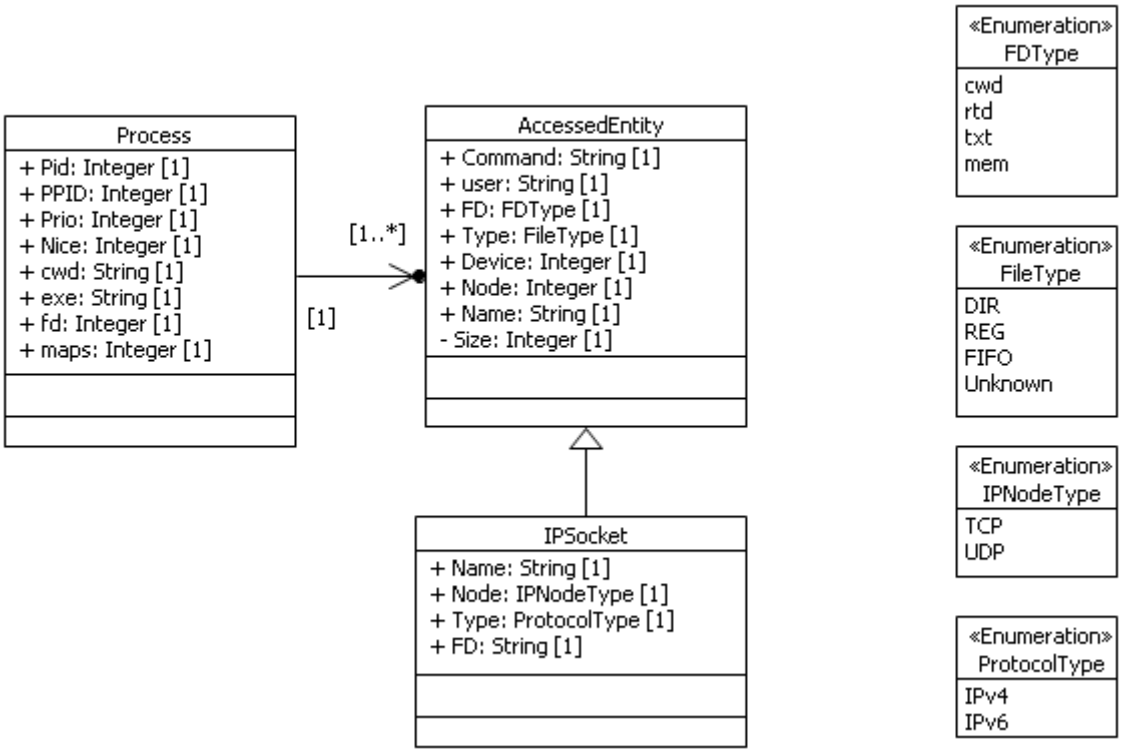
```
COMMAND     PID      USER   FD      TYPE     DEVICE     SIZE       NODE NAME
init          1      root  cwd       DIR      253,0     4096          2 /
init          1      root  rtd       DIR      253,0     4096          2 /
init          1      root  txt       REG      253,0    38652     362010 /sbin/init
init          1      root  mem       REG      253,0    93508    1411549
/lib/libselinux.so.1
init          1      root  mem       REG      253,0   245376    1411548
/lib/libsepol.so.1
init          1      root  mem       REG      253,0  1686224    1409047 /lib/libc-
2.5.so
/proc/4/exe
```

lsof -i

kimenet (részlet)

```
COMMAND     PID  USER   FD   TYPE DEVICE SIZE NODE NAME
dhclient   2627  root    4u  IPv4   9107       UDP *:68
portmap    2760   rpc    3u  IPv4   9442       UDP *:111
portmap    2760   rpc    4u  IPv4   9443       TCP *:111 (LISTEN)
rpc.statd  2795  root    3u  IPv4   9564       UDP *:854
rpc.statd  2795  root    6u  IPv4   9555       UDP *:851
rpc.statd  2795  root    7u  IPv4   9574       TCP *:857 (LISTEN)
hpiod      3012  root    0u  IPv4  11542       TCP localhost.localdomain:2208
(LISTEN)
python     3017  root    4u  IPv4  11562       TCP localhost.localdomain:2207
(LISTEN)
cupsd      3039  root    4u  IPv4  11662       TCP localhost.localdomain:631
(LISTEN)
cupsd      3039  root    6u  IPv4  11665       UDP *:631
sendmail   3061  root    4u  IPv4  11753       TCP localhost.localdomain:25
(LISTEN)
avahi-dae  3162 avahi   13u  IPv4  12230       UDP *:5353
avahi-dae  3162 avahi   14u  IPv6  12231       UDP *:5353
avahi-dae  3162 avahi   15u  IPv4  12232       UDP *:60195
avahi-dae  3162 avahi   16u  IPv6  12233       UDP *:54202
ekiga      6268  root   34u  IPv4  57799       TCP 192.168.80.146:1720 (LISTEN)
ekiga      6268  root   37u  IPv4  57803       UDP 192.168.80.146:5060
ekiga      6268  root   38u  IPv4  57804       UDP 192.168.80.255:5060
sshd      23495  root    3u  IPv6  64993       TCP *:22 (LISTEN)
```

2.1.3 Modell



1. ábra : Az lsof megfigyeléseinek modellje

2.1.4 Modell leírás

Process

- Pid : processz id
- Ppid : processz szülő id
- Prio : processz prioritás
- Nice : processzhez rendelt nice érték
- cwd : aktuális futási könyvtár
- exe : symlink a futtatott binárishoz
- fd : processz által megnyitott fileok leírójait tartalmazó alkönyvtár
- maps : aktzálisan felmappolt memóriaterületek és hozzáférési módusuk

Accessed entity

- Command: a futtatott parancs
- User : a futtatott parancs usere
- Fd : fájl leíró típusa
- Type : fájl típusa
- Device : a megnyitott fájlt tartalmazó eszköz
- Node : inode szám
- Name : elérési út és név
- Size : méret

IPSocket

Az Accessed entityből származó osztály, mely a megnyitott IPSocketeket reprezentálja.

- Name : IP cím és port
- Node : protokol típus [TCP|UDP]
- Type : IP protokoll típus [ipv4|ipv6]
- Megjegyzés
- Fd : értékei [x]u formátumúak lesznek ahol x egész szám

2.2 netstat

2.2.1 Eszköz leírása

A netstat eszköz segítségével információt nyerhetünk a rendszeren létező hálózati kapcsolatokról. Ezen kapcsolatok két típusra oszlanak, az egyik az internet kapcsolatok, a másik pedig a processzek közti kommunikációt lehetővé tevő Unix Domain Socketek. Ezen eszköz segítségével a processzek közötti függéseket vagyunk képesek feltésképezni egy hoszton belül, illetve, ha rendelkezésre áll egy olyan hoszt netstat kimenete, mellyel az kommunikál, úgy hoszok között is.

2.2.2 Használat

```
netstat -a -n -p
```

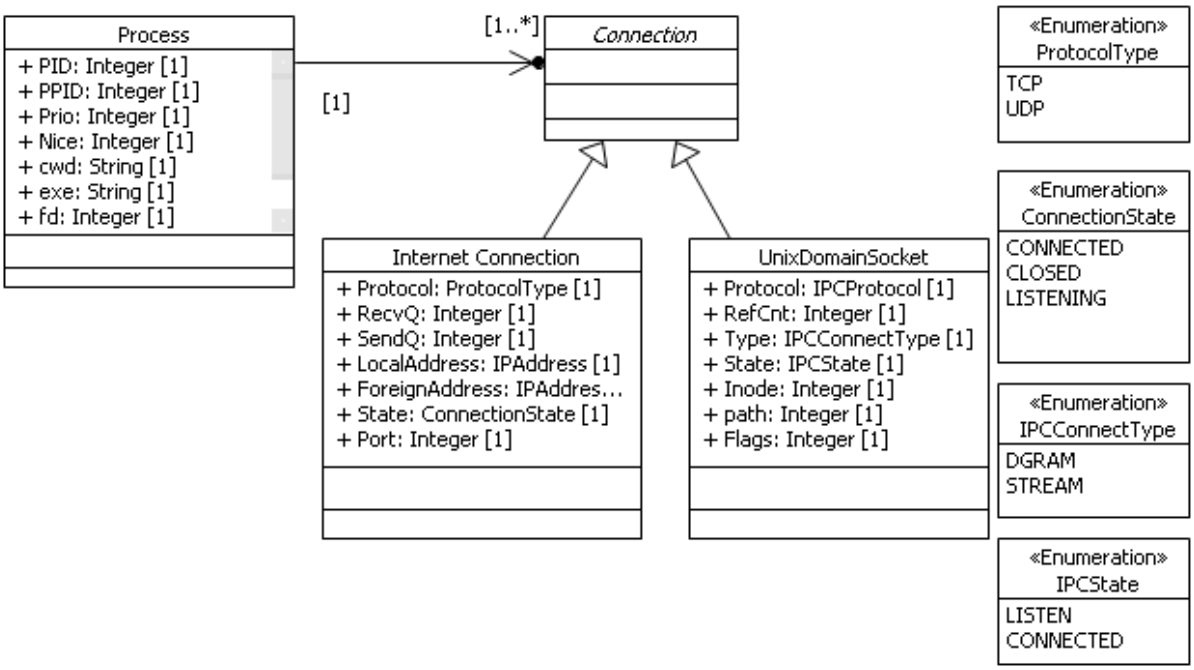
kimenet (részlet)

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
PID/Program name
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN
3012/hpiod
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
2760/portmap
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
3039/cupsd
tcp        0      0 192.168.80.146:1720         0.0.0.0:*                   LISTEN
6268/ekiga
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
3061/sendmail: acce
tcp        0      0 0.0.0.0:857                 0.0.0.0:*                   LISTEN
2795/rpc.statd
tcp        0      0 127.0.0.1:2207              0.0.0.0:*                   LISTEN
3017/python
tcp        0      0 :::22                       :::*                        LISTEN
23495/sshd
udp        0      0 0.0.0.0:854                 0.0.0.0:*
2795/rpc.statd
udp        0      0 0.0.0.0:5353                0.0.0.0:*
3162/avahi-daemon:
udp        0      0 0.0.0.0:111                 0.0.0.0:*
2760/portmap
udp        0      0 0.0.0.0:631                 0.0.0.0:*
3039/cupsd
udp        0      0 :::54202                    :::*

Active UNIX domain sockets (servers and established)
```

```
Proto RefCnt Flags       Type       State         I-Node PID/Program name    Path
unix  2      [ ACC ]     STREAM     LISTENING     14816  3542/mixer_applet2
/tmp/orbit-root/linc-dd6-0-1dbe6db95f12d
unix  2      [ ACC ]     STREAM     LISTENING     36815  6875/gedit
/tmp/gedit.root.1736556474
unix  2      [ ACC ]     STREAM     LISTENING     34746  5665/gnome-terminal
/tmp/orbit-root/linc-1621-0-159754989102e
unix  2      [ ACC ]     STREAM     LISTENING     14835  3544/notification-a
/tmp/orbit-root/linc-dd8-0-1dbe6db96758c
unix  2      [ ACC ]     STREAM     LISTENING     35465  6268/ekiga
/tmp/orbit-root/linc-187c-0-5fb5062956fe4
```

2.2.3 Modell



2. ábra : A netstat által feltérképezett környezet modellje

2.2.4 Modell leírás

Process

Mint az lsof nál lásd itt.

Connection

A két lehtséges kapcsolattípus ősosztályaként szolgáló absztrakt osztály.

Internet Connection

- Protocol : protokol típusa [TCP|UDP]
- RecvQ : az ehhez a socket-hez csatlakozott program által nem másolt byteok száma
- SendQ : a távoli hoszt által nem nyugtázott byteok száma

- LocalAddress : helyi cím
- ForeignAddress : távoli cím
- State : a kapcsolat állapota
- Port : port

UnixDomainSocket

- Protocol : a használt protokol, jelen esetben csak egy lehetséges értéket vehet fel: unix
- Refcnt : a socketre mutató referenciák száma
- Type : kapcsolat típusa [DGRAM|STREAM]
- State : IPC kapcsolat állapota
- Inode : inode szám
- Path : elérési út
- Flags : beállítható flagek

2.3 ps

2.3.1 Eszköz leírása

A ps paranccsal a rendszeren futó folyamatokról, és az őket indító parancsról (azaz binárisokról) kapunk információt.

2.3.2 Használat

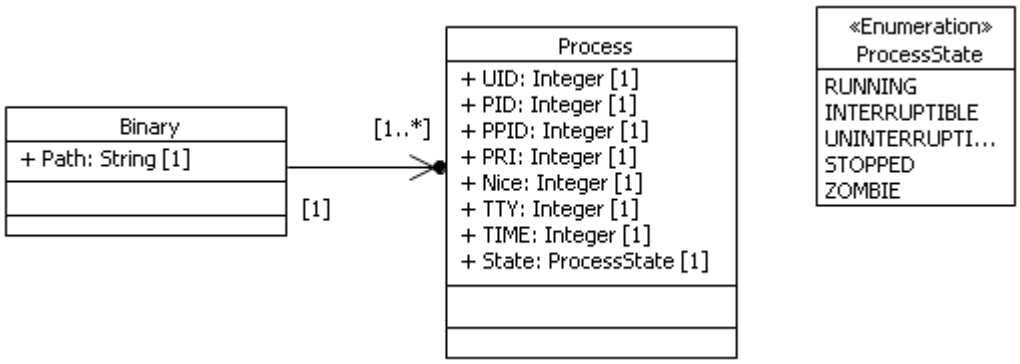
ps -e -all -f

kimenet (részlet)

```
F S UID        PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
4 S root         1     0  0  78   0 -   539 -      03:48 ?        00:00:02 init
[5]
1 S root         2     1  0 -40   - -     0 migrat 03:48 ?        00:00:01
[migration/0]
1 S root         3     1  0  99  19 -     0 ksofti 03:48 ?        00:00:00
[ksoftirqd/0]
5 S root         4     1  0 -40   - -     0 watchd 03:48 ?        00:00:00
[watchdog/0]
1 S root         5     1  0 -40   - -     0 migrat 03:48 ?        00:00:00
[migration/1]
1 S root         6     1  0  94  19 -     0 ksofti 03:48 ?        00:00:02
[ksoftirqd/1]
5 S root         7     1  0 -40   - -     0 watchd 03:48 ?        00:00:00
[watchdog/1]
1 S root         8     1  0  70  -5 -     0 worker 03:48 ?        00:00:00
[events/0]
1 S root         9     1  0  70  -5 -     0 worker 03:48 ?        00:00:00
[events/1]
1 S root        10     1  0  70  -5 -     0 worker 03:48 ?        00:00:00
[khelper]
5 S root        11     1  0  70  -5 -     0 worker 03:48 ?        00:00:00
[kthread]
1 S root        15    11  0  73  -5 -     0 worker 03:48 ?        00:00:00
[kblockd/0]
1 S root        16    11  0  70  -5 -     0 worker 03:48 ?        00:00:00
[kblockd/1]
1 S root        17    11  0  76  -5 -     0 worker 03:48 ?        00:00:00
[kacpid]
```

```
1 S root       182    11  0  71  -5 -     0 worker 03:48 ?        00:00:00
[cqueue/0]
1 S root       183    11  0  71  -5 -     0 worker 03:48 ?        00:00:00
[cqueue/1]
1 S root       186    11  0  70  -5 -     0 hub_th 03:48 ?        00:00:00 [khubd]
1 S root       188    11  0  70  -5 -     0 serio_ 03:48 ?        00:00:00
[kseriod]
1 S root       257    11  0  75   0 -     0 -      03:48 ?        00:00:00
[khungtaskd]
1 S root       258    11  0  75   0 -     0 pdflus 03:48 ?        00:00:00
[pdflush]
1 S root       259    11  0  75   0 -     0 pdflus 03:48 ?        00:00:03
[pdflush]
1 S root       260    11  0  70  -5 -     0 kswapd 03:48 ?        00:00:02
[kswapd0]
1 S root       261    11  0  71  -5 -     0 worker 03:48 ?        00:00:00 [aio/0]
1 S root       262    11  0  71  -5 -     0 worker 03:48 ?        00:00:00 [aio/1]
1 S root       481    11  0  71  -5 -     0 worker 03:48 ?        00:00:00
[kpsmoused]
1 S root       526    11  0  70  -5 -     0 worker 03:48 ?        00:00:00
[mpt_poll_0]
1 S root       527    11  0  74  -5 -     0 worker 03:48 ?        00:00:00 [mpt/0]
1 S root       528    11  0  74  -5 -     0 scsi_e 03:48 ?        00:00:00
[scsi_eh_0]
1 S root       532    11  0  73  -5 -     0 worker 03:48 ?        00:00:00 [ata/0]
1 S root       533    11  0  73  -5 -     0 worker 03:48 ?        00:00:00 [ata/1]
1 S root       534    11  0  73  -5 -     0 worker 03:48 ?        00:00:00
[ata_aux]
1 S root       541    11  0  74  -5 -     0 worker 03:48 ?        00:00:00
[kstriped]
```

2.3.3 Modell



3. ábra : A ps által felfedfezett összefüggések modellje

2.3.4 Modell leírás

Binary

Path : a bináris elérési útvonala

Process

- UID : felhasználó azonosító
- PID : processz azonosító
- PPID : processz ősének azonosítója
- PRI : processz prioritása
- Nice : processzhez rendelt nice érték
- TTY : a processz futási helye TTY[n] ahol n egész
- State : processz állapota

2.4 AppArmor

2.4.1 Eszköz leírása

Az AppArmor egy MAC (Mandatory Access Control) modul Linux kernelhez. Az AppArmor célja, hogy a rendszeren futó programok jogosultságait kezelje, biztosítva azt, hogy minden program csak a működéséhez szükséges erőforrásokhoz, fájlokhoz férhessen hozzá.

2.4.2 Működési leírás

Profilok

Az AppArmor által kontrollálni kívánt, elérési útjuk által azonosított programok mindegyikéhez egy-egy AppArmor profilt definiálunk. Ezen profilok tartalmazzák az adott programhoz rendelt engedélyeket és hozzáférési jogokat melyek a következő típusúak lehetnek:

- Capability

A Capability típusú szabályokkal definiálhatjuk, hogy az adott program milyen linux capability-vel legyen felruházva, például chown, fowner. A lehetséges értékek a függelék-ben találhatóak.

- Network

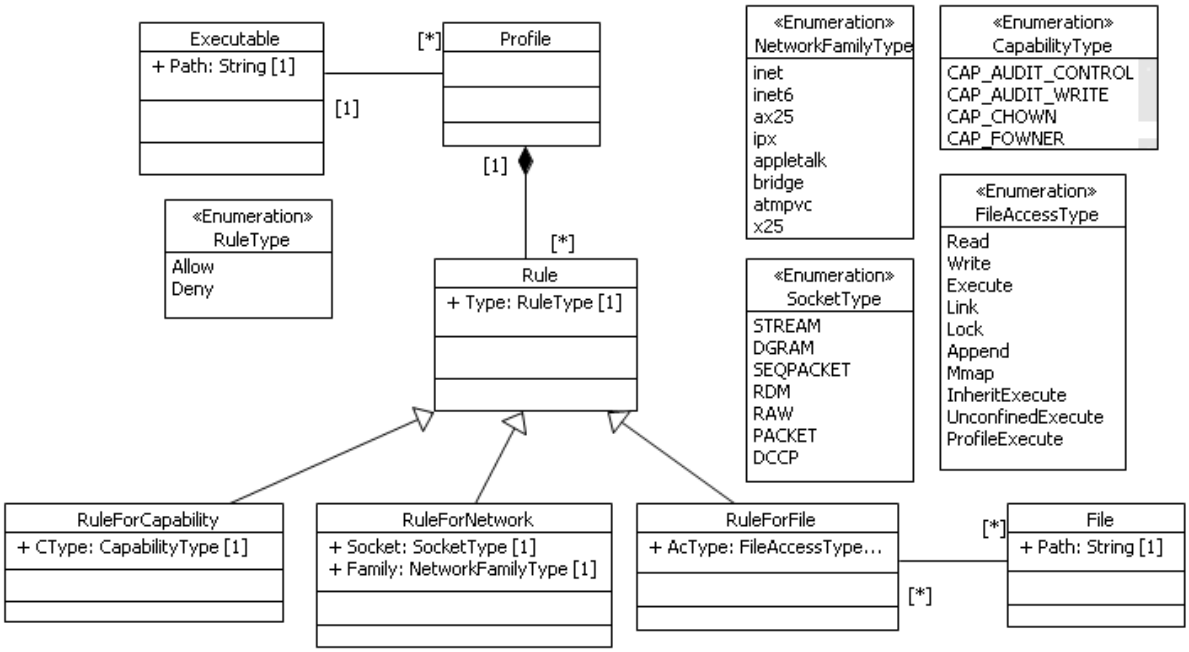
A Network típusú szabályokkal az irányított program hálózati használatát korlátozhatjuk: megadhatjuk az engedélyezett socketek típusait, valamint Network Family Type-ot.

- File/Directory

A fájl illetve könyvtár engedélyekkel az ezeken értelmezett klasszikus műveleteket (listázás, írás olvasás, végrehajtás) engedélyezhetjük vagy tilthatjuk igény szerint az eszköz segítségével.

Profilt két lényegesen különböző módon készíthetünk. Választhatjuk a manuális utat, mely során egy szövegfájlba az AppArmor által megkövetelt szintaktikát követve megírjuk a szabályokat. Ez a szabályok finomhangolását teszi lehetővé, ám a profilkészítési folyamat így nagyon lassú volna. A másik lehetőség, hogy az AppArmor *aa-genprof* eszközt használjuk. Az *aa-genprof* működés közben naplózza a megfigyelt program hozzáféréseit, majd ezekből alakítja ki a program profilját. Ez természetesen feltételezi azt, hogy a megfigyelt program a profil generálásakor nem tesz illegális műveletet.

2.4.3 Modell



4. ábra : AppArmor metamodellje

2.4.4 Modell leírás

Executable

- Path : a végrehajtható állomány elérési útja. Ezen elérési út azonosítja azt a végrehajtható fájlt amihez a profilt rendeljük.

Profile

Konténer osztály, mely a szabályokat tartalmazza. Explicit módon nincs neve, a végrehajtható file azonosítja.

Rule

- RuleType : Szabály típusa [Allow|Deny].

Az AppArmor alapvetően whitelist alapú engedélyezést használ, ám lehetőség van explicit módon tiltásokat definiálni.

RuleForCapability

- Ctype : Capability neve.

Ebben az esetben a megengedő illetve tiltó szabályokat úgy kell értelmezni, hogy az adott profillal ellátott végrehajtható file fel van-e ruházva a kérdéses capability-vel vagy sem

RuleForNetwork

- Socket : Socket típusa
- Family : hálózati család típusa

A hálózati szabályokkal azt határozhatjuk meg, hogy az adott processz milyen típusú hálózati kapcsolatokat használhat.

RuleForFile

- AcType : hozzáférés típusa

A fájl hozzáférési szabályokkla a fájlok és mappák elérését irányíthatjuk.

File

- Path : a szabályban szereplő fájl elérési útja.

2.4.5 Példa

Az aa-genprof segítségével a firefox programhoz generált profil.

```
#include <tunables/global>

/usr/lib/firefox/firefox.sh flags=(complain) {
  deny capability sys_ptrace,


  deny /bin/basename x,
  deny /bin/bash x,
  deny /bin/grep x,
  deny /etc/magic r,
  deny /usr/bin/file x,
  deny /usr/lib/firefox/firefox x,
  deny /usr/share/misc/magic.mgc r,

  owner /usr/lib/firefox/firefox.sh r,


  ^null-8b flags=(complain) {

    owner /root/.Xauthority r,
    owner /root/.adobe/Flash_Player/AssetCache/ r,
    owner /root/.macromedia/Flash_Player/#SharedObjects/ r,
    owner
/root/.macromedia/Flash_Player/macromedia.com/support/flashplayer/sys/settings.sol
r,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/_CACHE_001_ rw,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/_CACHE_002_ w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/_CACHE_003_ w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/content-prefs.sqlite k,
    owner /root/.mozilla/firefox/rbrnm5kh.default/cookies.sqlite-journal r,
    owner /root/.mozilla/firefox/rbrnm5kh.default/places.sqlite r,
    owner /root/.mozilla/firefox/rbrnm5kh.default/places.sqlite-journal w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/search.sqlite k,
```

```
    owner /root/.mozilla/firefox/rbrnm5kh.default/signons.sqlite k,
    owner /root/.mozilla/firefox/rbrnm5kh.default/urlclassifier3.sqlite k,

  }

  ^null-ca flags=(complain) {


    owner /proc/stat r,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/0C52A2D8d01 w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/26891F2Cd01 w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/52F9740Cd01 w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/_CACHE_001_ rw,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/_CACHE_002_ rw,
    owner /root/.mozilla/firefox/rbrnm5kh.default/Cache/_CACHE_003_ rw,
    owner /root/.mozilla/firefox/rbrnm5kh.default/XPC.mfasl r,
    owner /root/.mozilla/firefox/rbrnm5kh.default/cert8.db w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/content-prefs.sqlite rk,
    owner /root/.mozilla/firefox/rbrnm5kh.default/cookies.sqlite w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/cookies.sqlite-journal w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/key3.db w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/places.sqlite rwk,
    owner /root/.mozilla/firefox/rbrnm5kh.default/places.sqlite-journal rw,
    owner /root/.mozilla/firefox/rbrnm5kh.default/search.sqlite rk,
    owner /root/.mozilla/firefox/rbrnm5kh.default/sessionstore-1.js w,
    owner /root/.mozilla/firefox/rbrnm5kh.default/signons.sqlite rk,
    owner /root/.mozilla/firefox/rbrnm5kh.default/urlclassifier3.sqlite rk,
    owner /usr/lib/firefox/chrome/classic.jar r,
    owner /usr/lib/xulrunner-1.9.1.4/chrome/classic.jar r,
    owner /usr/lib/xulrunner-1.9.1.4/chrome/toolkit.jar r,
    owner /usr/share/icons/Oxygen_White/cursors/9d800788f1b08800ae810202380a0822
r,

  }
}
```

2.5 CoreForce

2.5.1 Az eszköz leírása

A CoreForce egy Windows rendeszrre íródott MAC/DAC rendszer. Célja, hogy az operációs rendszeren futó programok erőforrásokhoz való hozzáféréseit megfigyelje, illetve korlátozza, attól függően, hogy milyen módban fut.

2.5.2 Működési leírás

A CoreForce rendszer segítségével bármely végrehajtható fájlhoz létrehozhatunk egy hozzáférési profilt, mely ebben az esetben a következő elemekből áll.

Launch Control

A launch control tartalmazza az adott programhoz tartozó végrehajtható fileokat és azok szignatúráit. Azon végrehajtható fileok melyek itt nem szerepelnek de egy megfigyelt program mégis megpróbálja elidnítai, riasztást generálnak és beállítástól függően futásuk meghiúsulhat.

Security Level(s)

Minden megfigyelt programhoz lehetőségünk van több ún. biztonsági szintet megadni. A biztonsági szinteket a bennük alkalmazott szabályok engedékenysége alapján különböztethetjük meg.

Policy

Egy policy nem más mint hozzáférési szabályok gyűjteménye. Policy referálhat más policy-kra is, így lehetőség van akár hieralchikus módon felépíteni a hozzáférési szabályokat egy adott profilban. A policy jelentősége abban áll, hogy általa a léterhozott profilokban fellelhető hozzáférési szabályok újrahasznosíthatóvá válnak.

Permissions

A CoreForce rendszer alapvetően szintén whitelist alapú engedélyezést alkalmaz. Három típusú szabályt hozhatunk létre, egyrészről tűzfal beálítást, másrészt fájl illetve könyvtárhozzáférést, valamint registry key hozzzáférést.

- Tűzfal beállítások

Egy-egy szabállyal a tűzfalon megadhatjuk, az adott program által végezhető kommunikáció irányát, portját, protokoll típusát, helyi címét illetve távoli címét.

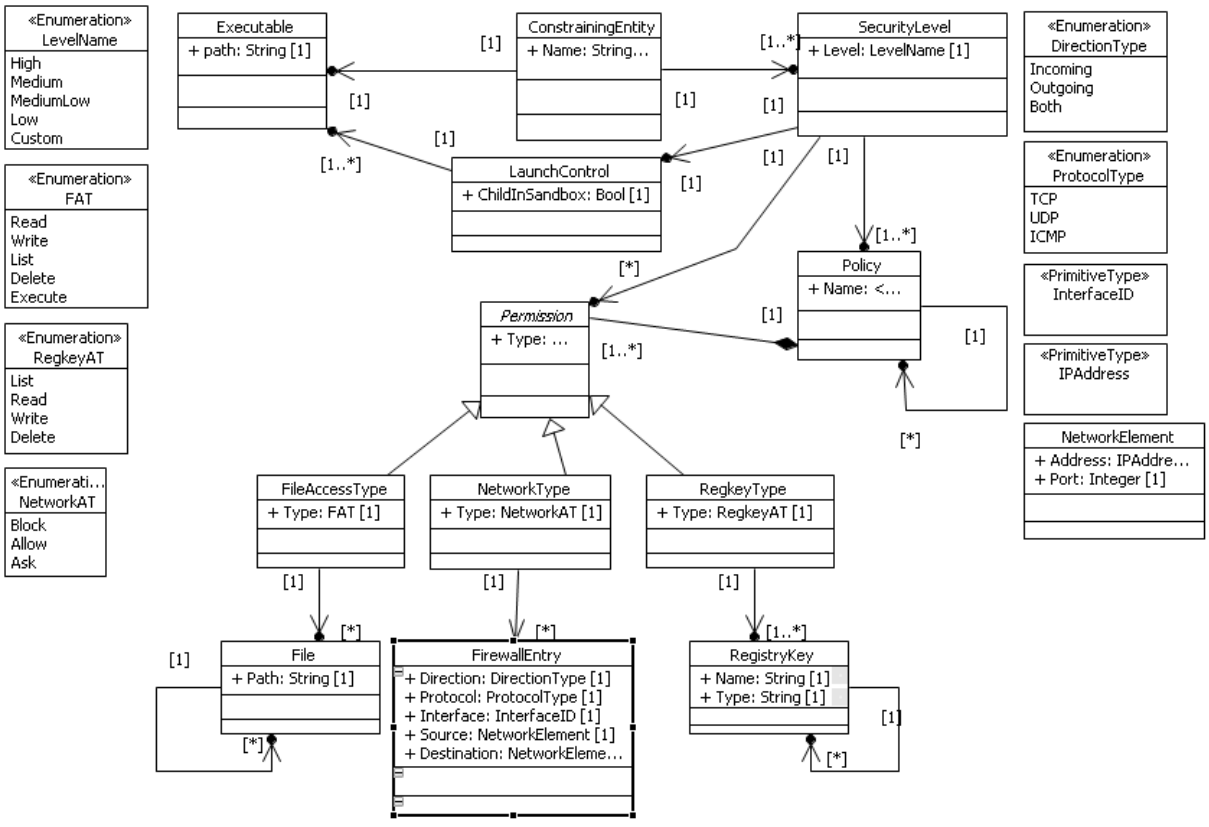
- Fájl és mappaelérések

A fájlok és mappák eléréseit a megszokott műveletek (olvasás, írás, listázás, végrehajtás, törlés) engedélyeinek beállításával végezhetjük. Lehetőségünk van az engedélyeket örököltetni, vagyis például egy mappa engedélyei érvényesek lehetnek a tartalmazott elemekre.

- RegistryKey elérések

Az ilyen típusú szabályok segítségével azt adhatjuk meg, hogy az adott program mely registry key-hez férhet hozzá. A megfigyelt programnak lehetősége lehet kilistázni, olvasni, írni illetve törölni az adott registry key-t

2.5.3 Modell



5. ábra : A CoreForce modellje

2.5.4 Modell leírás

Executable

- path : A megfigyelni kívánt program végrehajtható fájlának elérési útvonala

Security level

- level : A biztonsái szint neve

Launch Control

- ChildInSandbox : megadja, hogy a program által indított gyermek processzek ugyanabban a sandboxban lesznek-e mint a szűlőprogram, vagy sem.

Permission

Az engedélyek ősosztályaként szolgáló absztrakt osztály

FileAccessType

A fájlhozzáféréseket irányító típus

- Type : Az engedélyezett hozzáférés típusa

NetworkType

A hálózathasználatot irányító szabályok típusa

- Type : A hozzáférés típusa

RegkeyType

A registry key-ek hez tartozó enggedélyek típusa

- Type : a hozzáférés típusa

File

- Path : elérési út

FirewallEntry

Tűzfal bejegyzés, melyet engedélyezhetünk vagy tilthatunk.

- Direction : az engedélyezett forgalom iránya
- Protocol : az engedélyezett protokoll neve
- Interface : a kommunikáció hálózati interfésze
- Source : helyi cím
- Destionation : távoli cím

RegistryKey

A RegistryKey-eket reprezetnáló objektum

- Name : a registry key neve
- Type : a registry key típusa
- Value : a registty key értéke

2.5.5 Példa profil

A firefox programhoz létrehozott profilt a terjedelme miatt nem közlünk ebben a dokumentumban, ám mellékelve megtalálható coreforce_firefox_profile.cfx néven.

2.6 SELinux

2.6.1 Eszköz leírás

Az SELinux egy az amerikai NSA által fejlesztett linux alapú MAC rendszer. Megvalósítását tekintve az SELinux biztonsági irányultságú kernelmódosítások összessége, nem pedig önálló linux disztribúció. A hozzáférési szabályokat RBAC modell alapján végzi, azaz definiál szereplöröket és azok jogosultságait, majd ezeket a szerepköröket társítja processzekhez illetve felhasználókhoz. Az SELinux ugyanakkor a TE modell aspektusait is megvalósítja, mely a különböző processzek által elérni próbált objektumok típusai szerint engedélyez vagfy tilt bizonyos művelteket. Az eszköz megismerése során azt figyeltük meg, hogy az engedélyezések eldöntése leginkább az SELinux által az objektumokhoz rendelt típusok szerint történik. Fontos hangsúlyozni, hogy a fent említett típusok teljes mértékben függetlenek az objektum típusától mely lehet például fájl vagy socket.

2.6.2 Működési leírás

- Összefoglalva

Az SELinux rendszer először minden objektumhoz hozzárendel egy ún. kontextust, mely egy user:role:type hármas. A hozzáférési szabályokat a policy-k tartalmazzák, ahol egy szabály tipikusan a következőképp alakul: egy x kontextusban futó processz végrehajthat z műveletet egy y kontextusban lévő objektumon.

- A felcímkézés

Felcímkézésnek (labeling) azt a folyamatot nevezzük, mely során az SELinux a rendszerben található objektumokat kontextussal látja el. Egy SELinux rendszeren néhány, minden linuxon megtalálható parancs (ps, lsof, ls stb) speciális -Z paraméterrel kinyerhetőek az aktuális címkék.

Példa: ls / -Z

Kimenet

```
drwxr-xr-x  root root system_u:object_r:bin_t          bin
drwxr-xr-x  root root system_u:object_r:boot_t         boot
drwxr-xr-x  root root system_u:object_r:device_t       dev
drwxr-xr-x  root root system_u:object_r:etc_t          etc
drwxr-xr-x  root root system_u:object_r:home_root_t    home
```

```
drwxr-xr-x  root root system_u:object_r:lib_t          lib
drwx------  root root system_u:object_r:file_t         lost+found
drwxr-xr-x  root root system_u:object_r:mnt_t          media
drwxr-xr-x  root root system_u:object_r:autofs_t       misc
drwxr-xr-x  root root system_u:object_r:mnt_t          mnt
drwxr-xr-x  root root system_u:object_r:autofs_t       net
drwxr-xr-x  root root system_u:object_r:usr_t          opt
dr-xr-xr-x  root root system_u:object_r:proc_t         proc
drwxr-x---  root root root:object_r:user_home_dir_t    root
drwxr-xr-x  root root system_u:object_r:sbin_t         sbin
drwxr-xr-x  root root system_u:object_r:security_t     selinux
drwxr-xr-x  root root system_u:object_r:var_t          srv
drwxr-xr-x  root root system_u:object_r:sysfs_t        sys
drwxrwxrwt  root root system_u:object_r:tmp_t          tmp
drwxr-xr-x  root root system_u:object_r:usr_t          usr
drwxr-xr-x  root root system_u:object_r:var_t          var
```

Látható, hogy a kimenetben megjelent minden bejegyzéshez az user-role-type hármas. Bár messzemenő következtetéseket nem célszerű egyetlen kimentből levonni, megfigyelhetjük, hogy az user-role kettős majdnem minden elem esetében azonos. A félév során elvégzett vizsgálatok valamint egyéb az SELinux-al foglalkozó források [8] alapján mondhatjuk, hogy a gyakorlatban a szabályok érvényesítése leginkább az objeklumok SELinux által hozzárendelt típusai szerint történnek.

- Szabályok

Engedélyezési szabály : azt határozza meg, hogy egy x kontextusban futó processz egy y kontextusú obkektumon milyen műveleltet végezhet el.

Példa :

```
allow unconfined_t mytype_t:file read;
```

Azaz az unconfined_t típusú processznek a mytype_t típusú file objektumon olvasási jogot engedélyezünk.

Lehetőség van továbbá **típusátmeneti szabályok** alkotására objektumok és processzek számára is, ám mivel a függőségek felderítése szempontjából ezek a szabályok irrelevánsak, mélyebben nem foglalkoztunk velük.

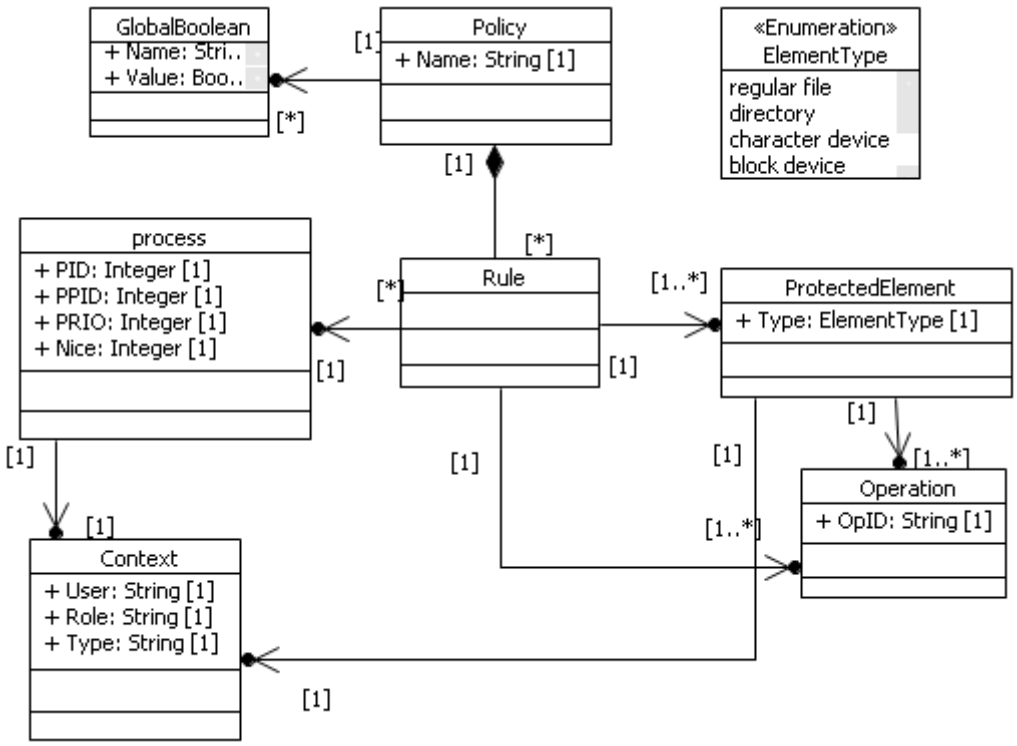
- Policy

A szabályokat egy ún policy fájlban definiáljuk, amely fájlt elkészültekor az SELinux-hoz erre a célra készült makefile segítségével le kell fordítani, hogy a rendszer a benne megfogalmazott engedélyezéseket fel tudja dolgozni.

- Booleans

A fentiekből kitűnik, hogy bármely változtatás hozzáférési jogosultságokban egy vagy akár több policy fájl módosítását is maga után vonhatja, ami nagyban megnehezíti a rendszert adminisztráló személyzet feladatát. Lehetőség van azonban a policy fájlok ujraforítása nélkül is némileg módosítani a rendszer működésén: az SELinux által használt boolean-ok állításával. Az egy adott rendszeren rendelkezésre álló booleanek listáját és aktuális értékeiket a *getsebool -a* paranccsal érhetjük el.

2.6.3 Modell



6. ábra : Az SELinux megfigyeléseinek modellje

2.6.4 Modell leírás

A modell megalkotásnál elsősorban azt vettük figyelembe, hogy az SELinux milyen megfigyeléseket képes tenni, kevésbé azt, hogy hogyan működik. A működési modell a függőségek felderítése szempontjából szükségtelenül nagy komplexitású lett volna.

Global Boolean

Az SELinux által használt booleanokat reprezentáló objektum

- Name : a boolena neve
- Value : értéke

Policy

A szabályokat tartalmazó konténer objektum.

- Name : a policy neve

Rule

A szabályokat reprezentáló asszociációs osztály, külön attribútuma nincs

Process

Mint az lsof nél.

ProtectedElement

Az SELinux nyilvántart elemtípusokat mint közönséges fájl, blokkos eszköz stb. Ezen elemekhez férhetnek hozzá a processzek

- Type : az objektum úgymond "valódi" típusa

Operation

Az engedélyezett művelet et reprezentáló objektum.

- OpID : az engedélyezett művelet neve.

Context

A különböző rendszerobjektumokhoz rendelt hármas, melyek alapján a policyben definiált szabályok érvényre jutnak.

- User : SELinux felhasználó. NB: nincs köze a rendszerbe bejelentkezett felhasználóhoz
- Role : SELinux role
- Type : SELinux által hozzárendelt típus

3 Függelék

3.1 Linux Capabilities

```
CAP_AUDIT_CONTROL (since Linux 2.6.11)
CAP_AUDIT_WRITE (since Linux 2.6.11)
CAP_CHOWN
CAP_DAC_OVERRIDE
CAP_DAC_READ_SEARCH
CAP_FOWNER
CAP_FSETID
CAP_IPC_LOCK
CAP_IPC_OWNER
CAP_KILL
CAP_LEASE
CAP_LINUX_IMMUTABLE
CAP_NET_ADMIN
CAP_NET_BIND_SERVICE
CAP_NET_BROADCAST
CAP_NET_RAW
CAP_SETGID
CAP_SETPCAP
CAP_SETUID
```

```
CAP_SYS_ADMIN
CAP_SYS_BOOT
CAP_SYS_CHROOT
CAP_SYS_MODULE
CAP_SYS_NICE
CAP_SYS_PACCT
CAP_SYS_PTRACE
CAP_SYS_RAWIO
CAP_SYS_RESOURCE
CAP_SYS_TIME
CAP_SYS_TTY_CONFIG
```

4 Hivatkozások

[1] "Constellation: automated discovery of service and host dependencies in networked systems" Paul Barham, Richard Black, Moises Goldszmidt, Rebecca Isaacs, John MacCormick, Richard Mortier, Aleksandr Simma Microsoft Research - TechReport, Citeseer -2008

[2] "Eureka : A Resource Discovery Service for Component Deployment" Karl Pauls, Richard S. Hall W. Emmerich and A.L. Wolf (Eds.): CD 2004, LNCS 3083, pp. 159-174, 2004. Springer-Verlag Berlin Heidelberg 2004

[3] "Core Force User's Guide" from http://corelabs.coresecurity.com/ visited on 01/10/2010

[4] "Managing networks through context: Graph visualization and exploration" QiLiao ,Andrew Blaich, Dirk Van Bruggen, Aaron Striegel Department of Computer Science and Engineering, University of Notre Dame, NotreDame, IN46556, UnitedStates , Computer Networks 2010

[5] "A Protection Mechanism for an Intrusion Detection System Based on Mandatory Access Control" Takefumi Onabuta, Tadashi Inoue, Midori Asaka Information-Technology Promotion Agency, Japan 2001

[6] "A survey of fault localization techniques in computer networks" Małgorzata Steinder, Adarshpal S.Sethi IBM T. J. Watson Research Center, Hawthorne, NY10532,USA Computer and Information Sciences, University of Delaware, Newark, DE, USA Received3October2003 accepted 7 January2004 Available online 1 July 2004

[7] "Designing role-based access control policies with UML" A. Cenys, A. Normantas* and L. Radvilavicius Information Security Laboratory, Department of Information System, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University Sauletekio al. 11, SRL-I-415, LT-10223, Vilnius, Lithuania Received 18 April 2009; Accepted 18 May 2009, Journal of Engineering Science and Technology Review 2 (1) (2009) p(48-50)

[8] "Breaking the Ice with SELinux" EliBillauer December 8th, 2008 obtained from http://www.billauer.co.il/lectures.html last visited: 08/10/2010