AppArmor

Roberto Cecchini

Corso di formazione sulla sicurezza informatica
Firenze 4-5 Novembre 2015

SELinux vs AppArmor

- SELinux
  - Sistema MAC
    - TE (Type Enforcement) con RBAC (Role Based Access Control) "general purpose";
    - MLS (Multi Level Security) sul modello BLP (Bell-La Padula) con l'estensione MCS (Multi-Category Security)
      - usata, ad es. nel caso di un file, per garantirne diversi livelli di segretezza all'interno una organizzazione
  - Default deny
  - Molto complesso
  - Overhead: 6% - 15%
  - Buona documentazione (anche perché è complicato...)

SeLinux vs AppArmor

- AppArmor:
  - Stabilisce cosa una singola applicazione può fare
  - Minimi privilegi, ma senza paradigmi globali di sicurezza
  - Default deny verso l'applicazione, ma ignora il resto del sistema
  - Relativamente user-friendly
  - Overhead < 2%
  - Documentazione migliorabile

SELinux vs AppArmor

- AA identifica gli oggetti via **path**
  - nuove installazioni, non tutto ne ha uno
- Il set di operazioni di AA è piccolo: **read**, **write**, **append**, **exec**, **lock** e **link**, in SEL anche **mknod**, **bind to socket**, **loading** e **unloading** di moduli kernel, **accessi alla memoria**, ecc. ecc.
- AA non supporta multi-level security
- SEL supporta un "remote policy server", le gestione centralizzata in AA è molto più complicata

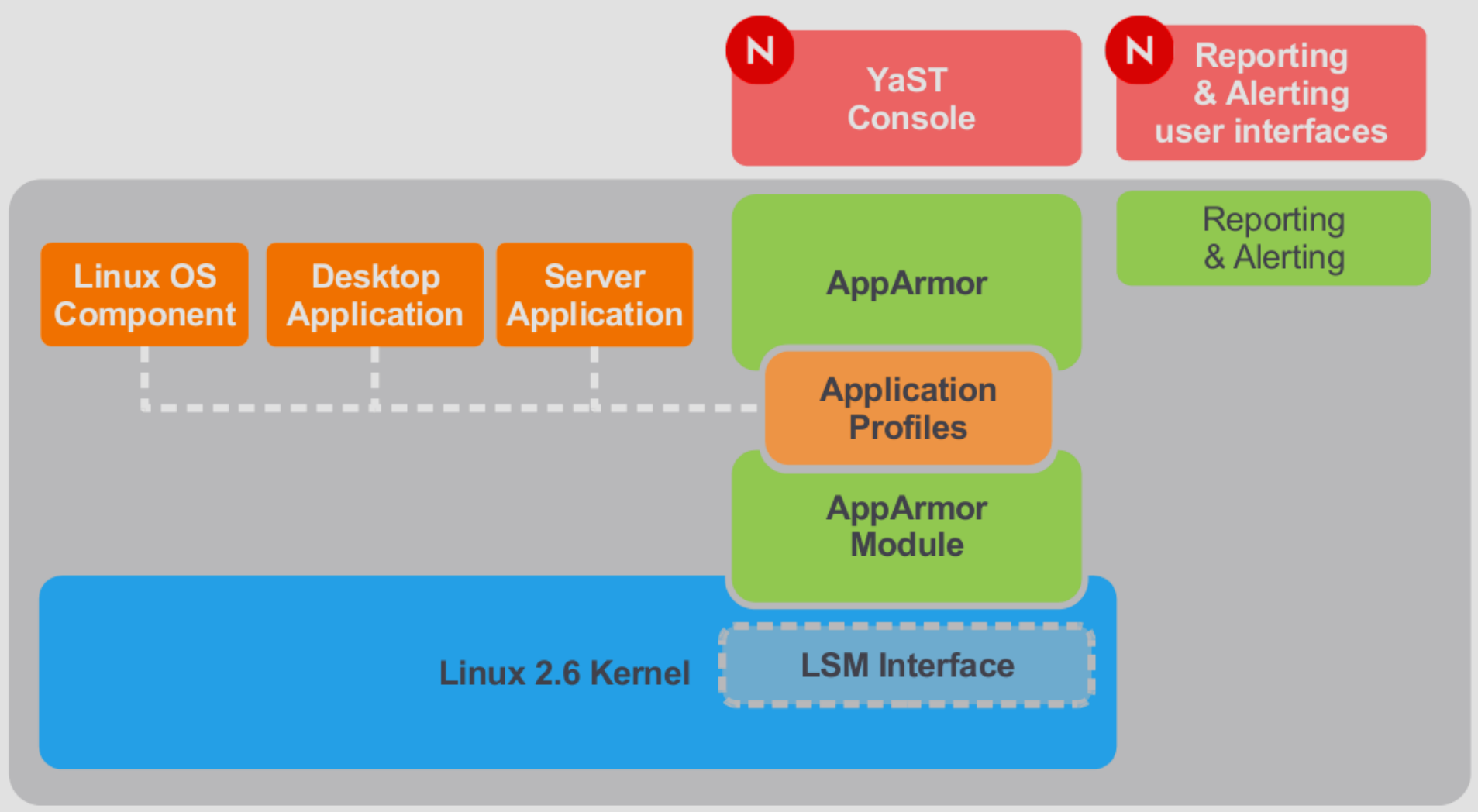
AppArmor

- Controlla le applicazioni tramite **profili**
- Libreria di profili per le applicazioni più comuni
  - “foundation classes” da includere
    - base, authentication, console, kerberos, nameservice, wutmp
- Strumenti per creare, modificare e controllare i profili
- Applicazioni modificate per permettere un controllo migliore dei sottoprocessi (ad es. Apache e Tomcat)
- **lkm** AppArmor

AppArmor: disponibilità

- Ultima versione (7/2015): 2.10
- Inclusa in:
  - Annvix
  - Arch Linux
  - Debian
  - Gentoo
  - Mandriva
  - openSUSE (integrata nell'installazione di default)
  - Pardus Linux
  - PLD
  - Ubuntu (integrata nell'installazione di default)

AppArmor e linux



Cosa (**non**) fa AppArmor

- I profili non sono user-specifici
- Meglio su sistemi con pochi user account
- Uso tipico:
  - programmi setuid o setgid;
  - programmi lanciati via cron;
  - applicazioni web (ad es. script cgi o pagine php);
  - applicazioni di rete.

AppArmor

- Modi di esecuzione (a livello di singolo profilo)
  - **`complain`**
  - **`enforce`**
- Controllo
  - **`rcapparmor start | stop | reload | status`**
- Profili
  - **`autodep`**
  - **`genprof`**
  - **`logprof`**
  - **`unconfined`**

rcapparmor status

```
apparmor module is loaded.
28 profiles are loaded.
28 profiles are in enforce mode.
   /sbin/klogd
   /sbin/syslog-ng
   /sbin/syslogd
   /usr/lib/apache2/mpm-prefork/apache2
   ...
   /usr/sbin/avahi-daemon
   /usr/sbin/dnsmasq
   /usr/sbin/dovecot
   /usr/sbin/identd
 ...
   /usr/sbin/ntpd
   /usr/sbin/smbd
   ...
   /usr/{sbin/traceroute,bin/traceroute.db}
  /{usr/,}bin/ping
0 profiles are in complain mode.
3 processes have profiles defined.
3 processes are in enforce mode.
   /usr/sbin/avahi-daemon (533)
   /usr/sbin/nscd (620)
   /usr/sbin/ntpd (1639)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

Messaggi di auditing

```
type=AVC msg=audit(1446195298.493:2240):
apparmor="DENIED" operation="open" parent=5588
profile="/usr/sbin/httpd2-prefork//DEFAULT_URI"
name="/srv/www/cgi-bin/" pid=5589 comm="ls"
requested_mask="r" denied_mask="r" fsuid=30 ouid=0

type=AVC msg=audit(1446195328.001:2241):
apparmor="DENIED" operation="exec" parent=5592
profile="/usr/sbin/httpd2-prefork//DEFAULT_URI"
name="/usr/bin/cat" pid=5593 comm="esegui"
requested_mask="x" denied_mask="x" fsuid=30 ouid=0
```

Per convertire le date

`date -d @1446195298`

Profili

- Profilo in modalità **complain**:
  - `sudo complain` /path/to/bin
- Profilo in modalità **enforce**:
  - `sudo enforce /path/to/bin`
- Ricarica un profilo:
  - `sudo apparmor_parser -r /etc/apparmor.d/<profile>`
- Disabilita un profilo:
  - `sudo disable` /path/to/bin

change_hat e **change_profile**

- Un'applicazione può modificare il profilo durante l'esecuzione
- **change_profile**: transizione a senso unico, tipicamente dopo inizializzazione per ridurre privilegi
- **change_hat**: dominuzione temporanea di privilegi, tipicamente **mod_perl** e **mod_php**

Creazione profili: **genprof**

- Execute access

```
Profile:  /usr/sbin/xinetd
Program:  xinetd
Execute:  /usr/lib/cups/daemon/cups-lpd
Severity: unknown

[(I)nherit] / (P)rofile / (U)nconfined / (D)eny / Abo(r)t / (F)inish
```

- File access

```
Profile:  /usr/sbin/httpd2-prefork
Path:     /etc/group
New Mode: r
[1 - #include <abstractions/nameservice>]
2 - /etc/group

[(A)llow] / (D)eny / (N)ew / (G)lob / Glob w/(E)xt / Abo(r)t / (F)inish
```

Esempio di profilo

```
#include <tunables/global> 1
/usr/bin/pip 2 { 3
   #include <abstractions/base> 4
   capability setgid, 5
   network inet tcp, 6
   link /etc/sysconfig/pip -> /etc/pip.conf, 7
   /bin/mount              ux,
   /dev/{,u}random 8       r,
   /etc/ld.so.cache        r,
   /etc/pip/*              r,
   /lib/ld-*.so*           mr,
   /lib/lib*.so*           mr,
   /proc/[0-9]**           r,
   /usr/lib/**             mr,
   /tmp/ 9                 r,
   /tmp/pip.pid            wr,
   /tmp/pip.*              lrw,
   /@{HOME}/.pip_file 10   rw,
   /@{HOME}/.pip_lock      kw,
   owner 11 /shared/pip/** rw,
   /usr/bin/pop            cx, 12
   /bin/**                 px -> bin_generic, 13

  profile /usr/bin/pop { 14
    /bin/bash             rmix,
    /bin/cat              rmix,
    /bin/more             rmix,
    /var/log/pop*         rwl,
    /etc/pop              r,
}

# pip's hat, pup.
   ^pup 15 {
       /lib/ld-*.so*      mr,
       /usr/bin/pup       px,
       /var/spool/*       rwl,
  }
}
```

1 This loads a file containing variable definitions.

2 The normalized path to the program that is confined.

3 The braces serve as a container.

4 Pulls in components of AppArmor profiles to simplify.

5 Enable each of the 29 POSIX.1e draft capabilities.

6 The kind of network access allowed to the application.

7 A rule specifying the source and the target of a link.

8 The braces make this rule apply to the path both with and without the content enclosed by the braces.

9 A path entry specifying what areas of the file system the program can access. The first part of a path entry specifies the absolute path of a file (including regular expression globbing) and the second part indicates permissible access modes. A whitespace of any kind can precede pathnames or separate the pathname from the access modes.

10 This variable expands to a value that can be changed without changing the entire profile.

11 An owner conditional rule, granting read and write permission on files owned by the user.

12 A transition to the local profile **/usr/bin/pop**.

13 A named profile transition to the profile **bin_generic** located in the global scope.

14 Local profile **/usr/bin/pop** definition.

15 A “hat” subprofile of the application.

RBAC

- **pam_apparmor** consente di confinare utenti in sottoprofili basati su **grupname**, **username** o un **profilo di default**
- **pam_apparmor** insieme a **capability** permette di garantire privilegi amministrativi a utenti non root

Tutorial (1)

```
sudo systemctl start apache2
sudo tail -f /var/log/audit/audit.log | grep apparmor
```

qualche comando nella pagina web

```
   grep Nov /var/log/access_log
   cat /etc/passwd
```

comandi vari apparmor

```
   sudo unconfined
   sudo rcapparmor status
sudo systemctl stop apache2
```

Tutorial (2)

```
sudo genprof /usr/sbin/httpd2-prefork
sudo systemctl start apache2
```

solo home page

```
sudo systemctl stop apache2
```

scrittura profilo

```
less /etc/apparmor.d/usr.sbin.httpd2-prefork
sudo rcapparmor status
```

verificare che **httpd2-prefork** sia in enforce mode

```
sudo tail -f /var/log/audit/audit.log | grep apparmor
```

qualche comando sulla home page

Profilo **apache** (1ª versione)

```
#include <tunables/global>
/usr/sbin/httpd2-prefork {
   #include <abstractions/apache2-common>
   #include <abstractions/base>
   #include <abstractions/nameservice>
   capability kill,
   capability net_bind_service,
   capability setgid,
   capability setuid,
   /etc/apache2/** r,
   /etc/mime.types r,
   /usr/lib{,32,64}/** mr,
   /usr/sbin/httpd2-prefork mr,
   /var/log/apache2/* w,
   /var/run/httpd2.pid rw,

  ^DEFAULT_URI {
      #include <abstractions/apache2-common>
      #include <abstractions/web-data>
      /var/log/apache2/access_log w,
  }
  ^HANDLING_UNTRUSTED_INPUT {
      #include <abstractions/apache2-common>
  }
}
```

apache **mod_apparmor**

- Ad ogni richiesta di URI
  - change_hat in **HANDLING_UNTRUSTED_INPUT** durante il parsing iniziale
  - dopo il parsing iniziale (in cascata)
    - change_hat nel corrispondente **AAHatName**
      - **AAHatName**: direttive <Directory>, <DirectoryMatch>, <Location> or <LocationMatch>
    - change_hat nell'URI
    - change_hat nel **AADefaultHatName**
      - **AADefaultHatName**: server o virtualhost
    - change_hat() in **DEFAULT_URI**
    - usa il profilo di default di apache

Tutorial (3)

sudo genprof /usr/sbin/httpd2-prefork

sudo systemctl start apache2

comandi “legittimi” sulla home

sudo systemctl stop apache2

aggiornamento profilo

less /etc/apparmor.d/usr.sbin.httpd2-prefork

sudo rcapparmor status

verificare che **httpd2-prefork** sia in enforce mode

sudo tail -f /var/log/audit/audit.log | grep apparmor

qualche comando sulla home page

Profilo **apache** (2ª versione)

```
#include <tunables/global>
/usr/sbin/httpd2-prefork {
  #include <abstractions/apache2-common>
  #include <abstractions/base>
  #include <abstractions/nameservice>
  capability  kill,
  capability  net_bind_service,
  capability  setgid,
  capability  setuid,
  /etc/apache2/** r,
  /etc/mime.types r,
  /usr/lib{,32,64}/** mr,
  /usr/sbin/httpd2-prefork mr,
  /var/log/apache2/* w,
  /var/run/httpd2.pid rw,

  ^DEFAULT_URI {
     #include <abstractions/apache2-common>
     #include <abstractions/base>
     #include <abstractions/perl>
     #include <abstractions/web-data>
     /srv/www/cgi-bin/esegui rix,
     /usr/bin/grep rix,
     /var/log/apache2/access_log rw,
     /var/log/apache2/error_log r,
  }

  ^HANDLING_UNTRUSTED_INPUT {
        #include <abstractions/apache2-common>
    }
}
```