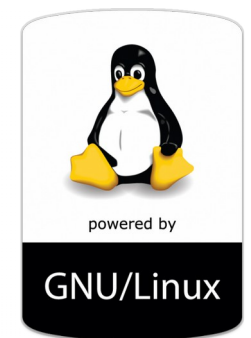




SELinux para todos



About Me

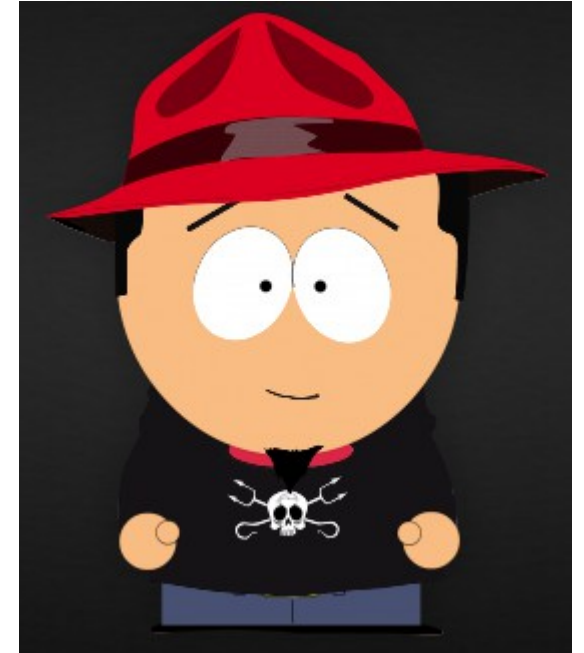
Alex Callejas

Technical Account Manager (Red Hat)

 dark_axl

 /rootzilopochtli

 www.rootzilopochtli.com



Geek by nature, Linux by choice, Fedora of course!

Que es SELinux?

- De donde vino?
 - Creado por la Agencia de Seguridad Nacional de los Estados Unidos (NSA) como un conjunto de parches para el kernel de Linux que utilizaba los Linux Security Modules (LSM)
 - Liberado por la NSA bajo la GNU Public License (GPL) en el año 2000
 - Adoptado por el kernel de Linux en 2003

Que es SELinux?

Es un ejemplo de Control de Acceso Mandatorio (*MAC: Mandatory Access Control*) en Linux



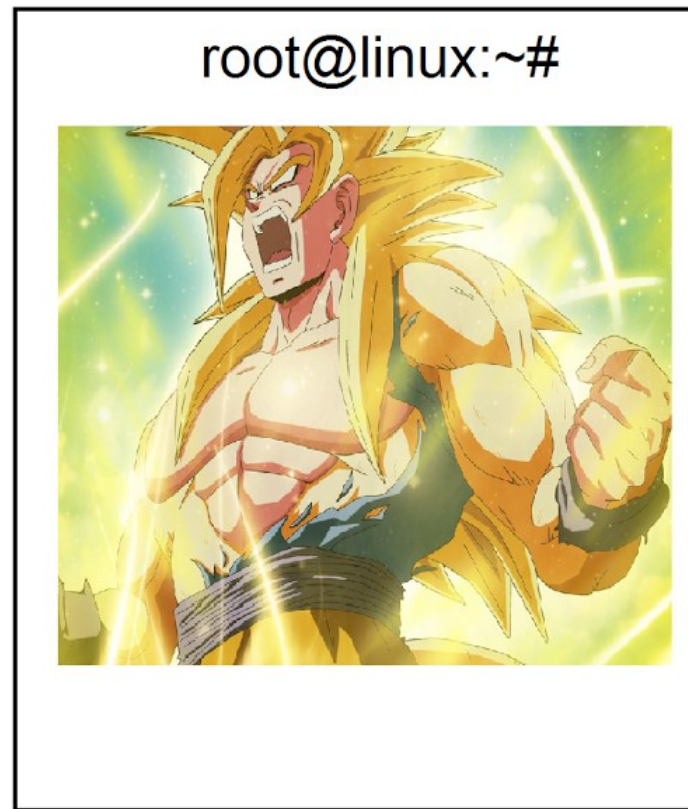
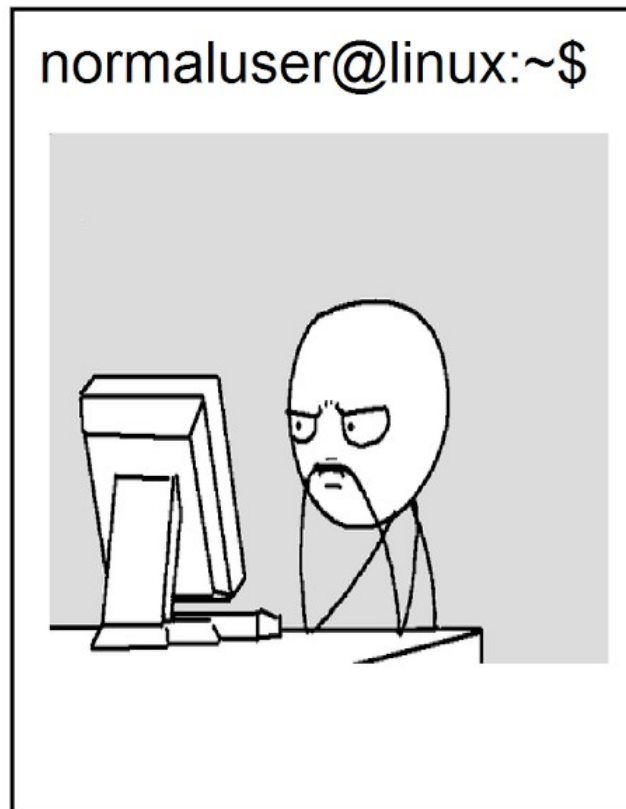
DAC vs MAC

- Históricamente, los sistemas Unix y Linux han utilizado el Control de Acceso Discrecional (DAC: Discretionary Access Control)
 - Propiedad (usuario, grupo, y otros) más permisos.
 - Los usuarios tienen la habilidad (a discreción) de cambiar permisos en sus archivos. Un usuario puede ejecutar: `chmod +rwx` en su directorio home, y nada puede detenerlo. Nada impedirá que otros usuarios o procesos accedan a los contenidos de su directorio home.

DAC vs MAC

- **root** es omnipotente

Differences between:



DAC vs MAC

- En un sistema con Control de Acceso Mandatorio, existen políticas que están fijas y configuradas.
- Aún si se cambia la configuración DAC en tu directorio home, si existe una política que impide que cualquier otro usuario o proceso lo accese, tu información esta segura.



DAC vs MAC

- Estas políticas pueden ser muy granulares, y determinar el acceso entre:
 - Usuarios
 - Archivos
 - Directorios
 - Memoria
 - Sockets
 - Puertos tcp/udp
 - etc...

Política

- La política por default es:
 - **targeted** - específica
 - Solo los procesos especificos (son cientos) son protegidos por SELinux
 - Todo lo demás se considera **no-confinado** (unconfined)
- Existe otro tipo:
 - **mls** -*multi-level/multi-category* security
 - Fuera del objetivo de la charla de hoy
 - Puede ser **muy** compleja
 - Utilizada por Agencias gubernamentales de tres siglas

Entonces, como funciona SELinux?

- Se puede determinar que política esta configurada en el sistema, verificando el archivo de configuración `/etc/selinux/config` (que además tiene una liga en `/etc/sysconfig/selinux`)

Ejecutando:

```
# getenforce  
  
# sestatus  
  
# cat /etc/selinux/config  
  
# cat /etc/sysconfig/selinux
```

Entonces, como funciona SELinux?

- Dos de los conceptos más importantes para entender SELinux son:
 - Labeling [etiquetado]
 - Type enforcement [tipo de ejecución]



Entonces, como funciona SELinux?

- **Labeling**

- Archivos, procesos, puertos, etc., son etiquetados con un contexto de SELinux
- Para los archivos y directorios, estas etiquetas son almacenadas como atributos extendidos en el file system
- Para los procesos, puertos, etc., el kernel administra las etiquetas

Entonces, como funciona SELinux?

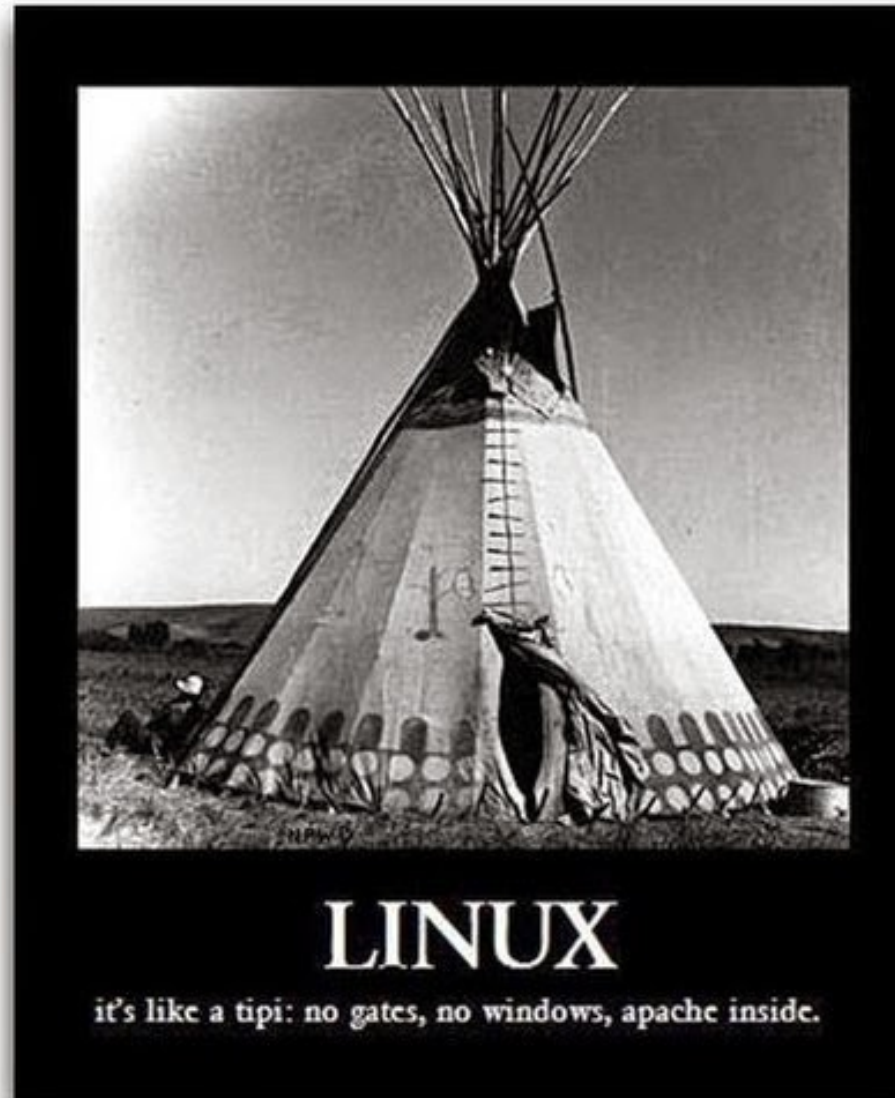
- Las etiquetas tienen el formato:
 - `user:role:type:level(optional)`
- En esta presentación trabajaremos únicamente con `type`, ya que `user`, `role` y `level` son utilizadas en implementaciones muy avanzadas con SELinux (MLS/MCS)

Entonces, como funciona SELinux?

- Como ejemplo demostrativo, veremos un servicio bastante complejo, uno que proporciona acceso desde la red, potencialmente en varios puertos, y potencialmente, da acceso a todo nuestro sistema e información.

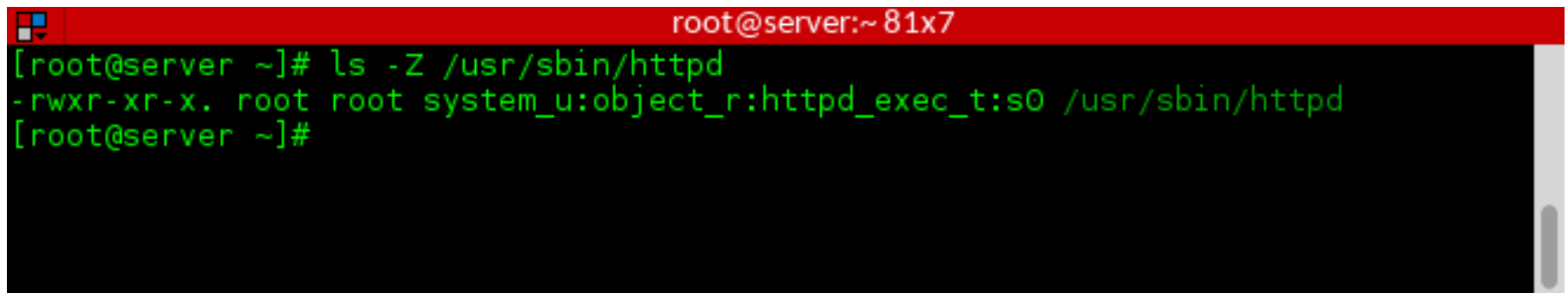


Entonces, como funciona SELinux?



Entonces, como funciona SELinux?

- El servidor web **Apache** no es necesariamente inseguro, sólo es que su rango de acceso es muy amplio.
- Tiene un binario ejecutable que se lanza desde `/usr/sbin`. Cuando observamos el contexto de SELinux de ese archivo, encontramos que su tipo es `httpd_exec_t`

A terminal window with a red title bar containing the text 'root@server:~ 81x7'. The terminal content shows a command being executed: '[root@server ~]# ls -Z /usr/sbin/httpd'. The output is: '-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd'. The prompt returns to '[root@server ~]#'.

```
root@server:~ 81x7
[root@server ~]# ls -Z /usr/sbin/httpd
-rwxr-xr-x. root root system_u:object_r:httpd_exec_t:s0 /usr/sbin/httpd
[root@server ~]#
```

Entonces, como funciona SELinux?

- El directorio de configuración del web server esta *etiquetado* como: `httpd_config_t`

```
root@server:~ 81x7
[root@server ~]# ls -dZ /etc/httpd
drwxr-xr-x. root root system_u:object_r:httpd_config_t:s0 /etc/httpd
[root@server ~]#
```

Entonces, como funciona SELinux?

- El directorio de logs del web server esta *etiquetado* como: `httpd_log_t`

```
root@server:~ 81x7
[root@server ~]# ls -dZ /var/log/httpd/
drwx----- . root root system_u:object_r:httpd_log_t:s0 /var/log/httpd/
[root@server ~]#
```

Entonces, como funciona SELinux?

- El directorio de contenido del web server esta *etiquetado* como: `httpd_sys_content_t`

```
root@server:~ 81x7
[root@server ~]# ls -dZ /var/www/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/
[root@server ~]#
```

Entonces, como funciona SELinux?

- El script de arranque del web server esta *etiquetado* como: `httpd_unit_file_t`

```
root@server:~ 103x7
[root@server ~]# ls -lZ /usr/lib/systemd/system/httpd.service
-rw-r--r--. root root system_u:object_r:httpd_unit_file_t:s0 /usr/lib/systemd/system/httpd.service
[root@server ~]#
```


Entonces, como funciona SELinux?

- Cuando el web server esta ejecutándose, el proceso se *etiqueta* como: httpd_t

```
root@server:~ 132x12
[root@server ~]# ps auxfZ | grep httpd
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 root 1373 0.0  0.0 113000 2108 pts/0 S+ 02:31  0:00      \_ grep --color=
auto httpd
system_u:system_r:httpd_t:s0      root      1354  0.1  0.5 360192 23080 ?          Ss  02:30  0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    1355  0.0  0.3 362276 14156 ?          S   02:30  0:00 \_ /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    1356  0.0  0.3 362276 14156 ?          S   02:30  0:00 \_ /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    1357  0.0  0.3 362276 14156 ?          S   02:30  0:00 \_ /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    1358  0.0  0.3 362276 14156 ?          S   02:30  0:00 \_ /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0      apache    1359  0.0  0.3 362276 14156 ?          S   02:30  0:00 \_ /usr/sbin/httpd -DFOREGROUND
[root@server ~]#
```

Entonces, como funciona SELinux?

- Si observamos los puertos en los que escucha el servidor web, veremos que incluso ellos están etiquetados

```
root@server:~ 137x16
[root@server ~]# netstat -tnlpZ | grep httpd
tcp6      0      0 :::80          :::*           LISTEN        3243/httpd    system_u:system_r:httpd_t:s0
tcp6      0      0 :::443         :::*           LISTEN        3243/httpd    system_u:system_r:httpd_t:s0

[root@server ~]# semanage port -l | grep http
http_cache_port_t    tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t    udp      3130
http_port_t          tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t  tcp      5988
pegasus_https_port_t tcp      5989
[root@server ~]#
```

Entonces, como funciona SELinux?

- Ahora bien... el archivo `/etc/shadow` tendrá una etiqueta `shadow_t`

```
root@server:~ 75x8
[root@server ~]# ls -lZ /etc/shadow
------. root root system_u:object_r:shadow_t:s0 /etc/shadow
[root@server ~]#
```

Entonces, como funciona SELinux?

- **Type enforcement**

- Bajo estos mismos conceptos, hace sentido que un proceso corriendo en el contexto `httpd_t` interactúe con un archivo etiquetado como `httpd_config_t`
- ¿Será la misma situación cuando un proceso corriendo en el contexto `httpd_t` interactúe con un archivo etiquetado como, digamos, `shadow_t`, sería correcto?

Entonces, como funciona SELinux?

- El tipo de ejecución (**type enforcement**) es la parte de la política que dice, por cada instancia: “*un proceso corriendo con la etiqueta `httpd_t` puede tener acceso de lectura a un archivo etiquetado como `httpd_config_t`*”

Como lidiar con las etiquetas?

- Utilizamos el argumento **-Z** en diferentes comandos para revisar el contexto, muchos de ellos lo aceptan:
 - `ls -Z`
 - `id -Z`
 - `ps -Z`
 - `netstat -Z`

Como lidiar con las etiquetas?

- El mismo argumento `-Z` lo podemos utilizar para crear y modificar archivos y contextos
 - `cp -Z`
 - `mkdir -Z`

Como lidiar con las etiquetas?

- También se pueden utilizar las herramientas de SELinux: `chcon` o `restorecon` para modificar los contextos de un archivo (lo veremos más adelante)
- Los contextos son configurados cuando los archivos son creados, basados en el contexto de su directorio padre (con algunas excepciones)
- Los RPM's pueden configurar contextos como parte de su instalación
- El proceso de login asigna el contexto por default (`unconfined` en la política `targeted`)

Como lidiar con las etiquetas?

- Transición de archivos (definidos por la política)
 - Si una aplicación `foo_t`, crea un archivo en un directorio etiquetado como `bar_t`, es posible que la política requiera una transición, entonces el archivo se crea con la etiqueta `baz_t`
 - Por Ejemplo: Un proceso, **dhclient**, ejecutándose con la etiqueta `dhclient_t`, crea un archivo `/etc/resolv.conf`, etiquetado como `net_conf_t`, en un directorio, `/etc`, con la etiqueta `etc_t`. Sin la transición, `/etc/resolv.conf` podría heredar la etiqueta `etc_t`.

Como lidiar con las etiquetas?

- También utilizamos el comando `semanage`, que puede administrar la configuración de SELinux:
 - Login
 - user
 - port
 - interface
 - Module
 - node
 - File context
 - Boolean
 - Estado Permitivo
 - dontaudit

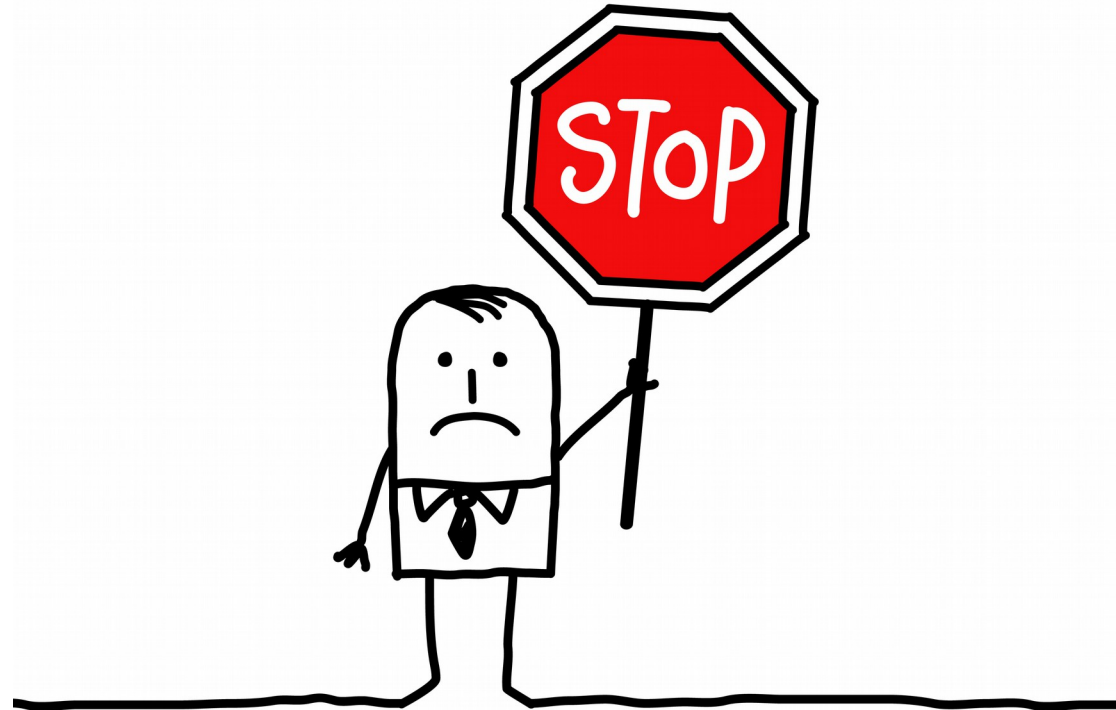
Y si SELinux me manda un mensaje de error?



VG24/7

NO DESHABILITES SELINUX!!!

- Deshabilitar SELinux es como subirle al estereo al máximo cuando escuchamos un ruido extraño en nuestro auto.



stopdisablinglinux.com

Seriously, stop disabling SELinux.

[Learn how to use it](#) before you blindly shut it off.

**Every time you run `setenforce 0`, you make [Dan Walsh](#) weep.
Dan is a nice guy and he certainly doesn't deserve that.**

Que trata de decirme SELinux?

- Existen 4 causas principales de errores en SELinux:
 - 1) Etiquetas (SELinux==Labeling)
 - 2) SELinux necesita saber
 - 3) La Aplicación/Política de SELinux puede tener bugs
 - 4) Tu información puede estar COMPROMETIDA!!!

Etiquetado (SELinux==Labeling)

- Cada proceso y objeto en el sistema tiene una etiqueta asociada con él
- Si tus archivos **no están etiquetados correctamente** el acceso puede ser negado
- Si utilizas **rutas alternas/personalizadas** para dominios confinados, *SELinux necesita saber*
 - Archivos **http** en `/srv/myweb` en lugar de `/var/www/html`?
 - Dile a SELinux!

```
root@server:~ 86x7
[root@server ~]# semanage fcontext -a -t httpd_sys_content_t '/srv/myweb(/.*)?'
[root@server ~]#
[root@server ~]# restorecon -R /srv/myweb
[root@server ~]#
```

Etiquetado equivalente

```
root@server:~ 86x3  
[root@server ~]# semanage fcontext -a -e /srv/myweb /var/www
```

- Este comando le dice a SELinux que todos los archivos bajo `/srv/myweb` son similares a `/var/www`
 - Por lo tanto: `/srv/myweb/cgi-bin/mycgi.cgi` será etiquetado como `httpd_sys_script_t`
- Para etiquetar todos los archivos bajo `/export/home` como si estuvieran bajo `/home`
 - `/export/home/flisol/.ssh` será etiquetado como `ssh_home_t`

```
root@server:~ 83x3  
[root@server ~]# semanage fcontext -a -e /export/home /home  
[root@server ~]#
```

SELinux necesita saber

- ¿Como configuraste tu apache server?
 - Dile a SELinux!!
- Si quieres que `httpd` envíe correo

```
# setsebool -P httpd_can_sendmail 1
```
- Configuración de `vsftp` en el login de usuarios

```
# setsebool -P ftp_home_dir 1
```
- HTTPD va a escuchar en el puerto 8585

```
# semanage port -a -t http_port_t -p tcp 8585
```

Qué son los Booleanos?

- Los Booleanos son configuraciones simples de SELinux que se pueden encender/apagar
 - Desde algo muy sencillo como “*permitamos que el servidor ftp accese a los directorios home*”, a cuestiones más esótericas como “*httpd puede utilizar mod_auth_ntlm_winbind**”

* `mod_auth_ntlm_winbind` is a pretty cool Apache module that will do authentication against Active Directory with NTLM (http://adldap.sourceforge.net/wiki/doku.php?id=mod_auth_ntlm_winbind).

Qué son los Booleanos?

- Para ver todos los booleanos, ejecutamos:

```
root@server:~ 76x29
[root@server ~]# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
auditadm_exec_content --> on
authlogin_nsswitch_use_ldap --> off
authlogin_radius --> off
authlogin_yubikey --> off
awstats_purge_apache_log_files --> off
boinc_execmem --> on
cdrecord_read_content --> off
cluster_can_network_connect --> off
cluster_manage_all_files --> off
cluster_use_execmem --> off
cobbler_anon_write --> off
cobbler_can_network_connect --> off
cobbler_use_cifs --> off
cobbler_use_nfs --> off
collectd_tcp_network_connect --> off
condor_tcp_network_connect --> off
conman_can_network --> off
cron_can_relabel --> off
cron_userdomain_transition --> on
cups_execmem --> off
cvs_read_shadow --> off
daemons_dump_core --> off
daemons_enable_cluster_mode --> off
```

Qué son los Booleanos?

- Para ver la explicación de cada uno, ejecutamos:

```
root@server:~ 112x29

[root@server ~]# semanage boolean -l
SELinux boolean      State  Default Description
ftp_home_dir         (off  ,  off)  Allow ftp to home dir
smartmon_3ware       (off  ,  off)  Allow smartmon to 3ware
mpd_enable_homedirs  (off  ,  off)  Allow mpd to enable homedirs
xdm_sysadm_login     (off  ,  off)  Allow xdm to sysadm login
xen_use_nfs          (off  ,  off)  Allow xen to use nfs
mozilla_read_content (off  ,  off)  Allow mozilla to read content
ssh_chroot_rw_homedirs (off ,  off) Allow ssh to chroot rw homedirs
mount_anyfile        (on   ,  on)   Allow mount to anyfile
cron_userdomain_transition (on ,  on)   Allow cron to userdomain transition
icecast_use_any_tcp_ports (off ,  off) Allow icecast to use any tcp ports
openvpn_can_network_connect (on ,  on)   Allow openvpn to can network connect
zoneminder_anon_write (off ,  off) Allow zoneminder to anon write
minidlna_read_generic_user_content (off ,  off) Allow minidlna to read generic user content
spamassassin_can_network (off ,  off) Allow spamassassin to can network
gluster_anon_write   (off ,  off) Allow gluster to anon write
deny_ptrace          (off ,  off) Allow deny to ptrace
selinuxuser_execmod (on   ,  on)   Allow selinuxuser to execmod
httpd_can_network_relay (off ,  off) Allow httpd to can network relay
openvpn_enable_homedirs (on   ,  on)   Allow openvpn to enable homedirs
glance_use_execmem   (off ,  off) Allow glance to use execmem
telepathy_tcp_connect_generic_network_ports (on ,  on) Allow telepathy to tcp connect generic network ports
httpd_can_connect_mythtv (off ,  off) Allow httpd to can connect mythtv
unconfined_mozilla_plugin_transition (on ,  on) Allow unconfined to mozilla plugin transition
sasauthd_read_shadow (off ,  off) Allow sasauthd to read shadow
```


Qué son los Booleanos?

- Para configurar un booleano, ejecutamos

```
# setsebool [booleano] [0|1]
```

- Para hacerlo permanente agregamos el argumento `-P`

```
root@server:~ 78x9
[root@server ~]# setsebool httpd_enable_ftp_server 1 -P
[root@server ~]#
[root@server ~]# getsebool httpd_enable_ftp_server
httpd_enable_ftp_server --> on
[root@server ~]#
```

La App/Policy puede tener bugs

- La política de SELinux puede tener bugs
 - Rutas inusuales en el código
 - Configuraciones
 - Redirección del stdout

La App/Policy puede tener bugs

- La Aplicación puede tener bugs
 - File descriptors filtrados
 - Memoria ejecutable
 - Librerías mal contruídas

- Reporta los bugs en Bugzilla para que los podamos arreglar

Tu información puede estar **COMPROMETIDA!!!**

- Si las herramientas actuales no hacen un buen trabajo al diferenciar contextos
- Si tienes dominios confinados que intentan:
 - Cargar módulos de kernel
 - Apagar el modo enforcing de SELinux
 - Escribir a `etc_t/shadow_t`
 - Modificar reglas de `iptables`
- **Tu información puede estar en PELIGRO**

Tips

- Instala setroubleshoot y setroubleshoot-server en tus equipos. Ellos tienen muchas herramientas que te ayudarán a diagnosticar y corregir problemas con SELinux
- Reinicia el servicio audit después de instalarlo

```
[root@server ~]# yum -y install setroubleshoot setroubleshoot-server
```

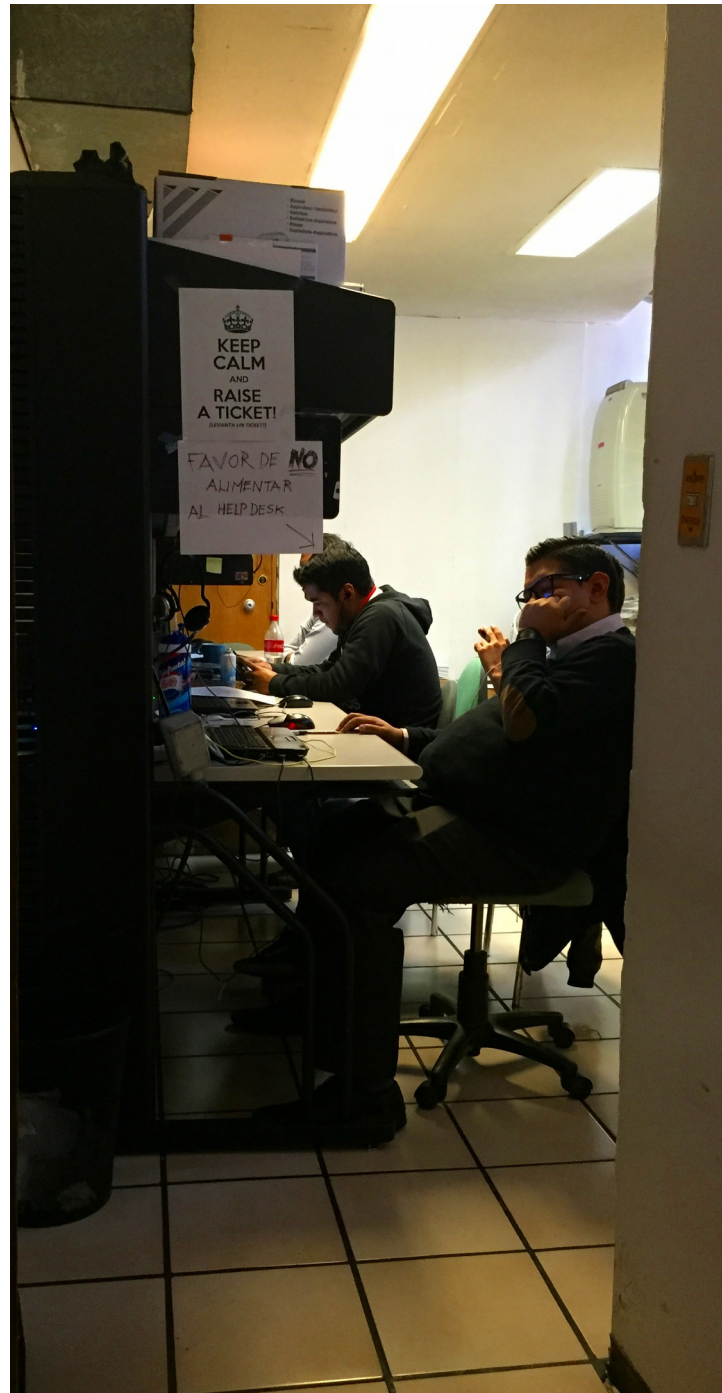
```
[root@server ~]# service auditd restart
```

```
Stopping logging: [ OK ]
```

```
Redirecting start to /bin/systemctl start auditd.service
```

```
[root@server ~]#
```

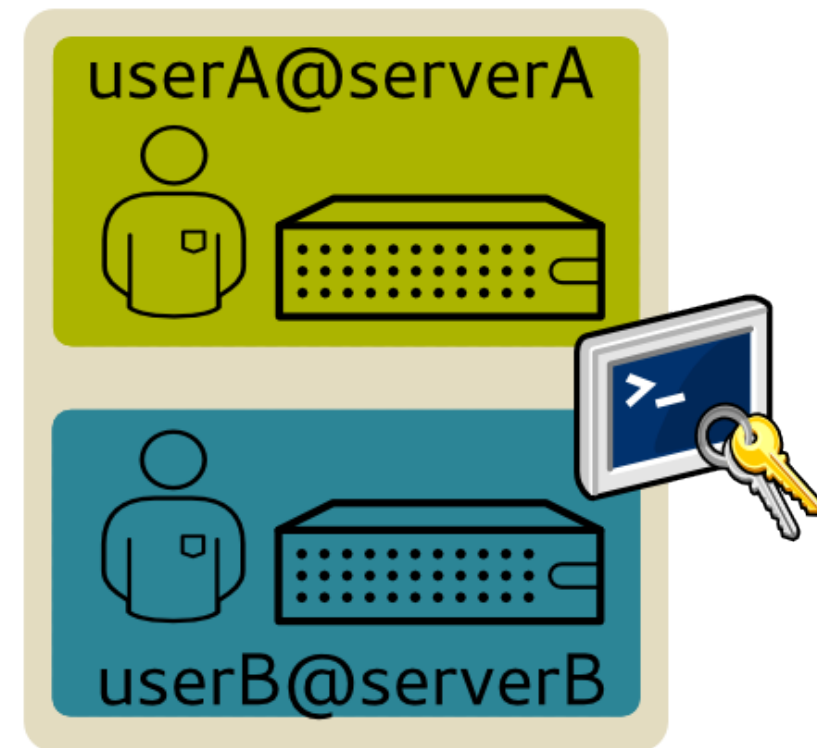
En la vida real



SELinux: Caso de la vida real

<http://www.rootzilopochtli.com/2014/12/que-hacer-cuando-algo-falla-ep2/>

- Problema: Mecánismo de confianza con llaves de SSH no funciona
- Análisis preliminar
 - Revisión de llave
 - Permisos
 - Análisis de logs
 - Regenerar mecanismo



SELinux: Caso de la vida real

- Análisis de Seguridad
 - Revisión de enforcement
 - Revisión de contextos

```
[root@serverA ~]# getenforce
```

```
Enforcing
```

```
[root@serverA ~]# ls -lZd /home/userA/
```

```
drwx-----. userA userA unconfined_u:object_r:user_home_dir_t:s0 /home/userA/
```

```
[root@serverA ~]#
```

```
[root@clientB ~]# getenforce
```

```
Enforcing
```

```
[root@clientB ~]# ls -lZd /opt/userB/
```

```
drwx-----. userB userB system_u:object_r:file_t:s0 /opt/userB/
```

```
[root@clientB ~]#
```


SELinux: Caso de la vida real

- Análisis de Seguridad
 - Instalar setroubleshoot
 - Revisando logs

```
[root@clientB ~]# tail /var/log/messages
```

```
Dec 12 16:56:32 clientB kernel: type=1305  
audit(1418424992.590:4): audit_pid=952 old=0  
auid=4294967295 ses=4294967295  
subj=system_u:system_r:auditd_t:s0 res=1
```

```
Dec 12 16:58:26 clientB setroubleshoot: SELinux  
is preventing /usr/sbin/sshd from search access  
on the directory /opt/userB. For complete SELinux  
messages. run sealert -l f963dd2e-ab81-4c3a-99ad-  
4d4d1e6736d5
```

```
[root@clientB ~]#
```

SELinux: Caso de la vida real

- Ejecutando sealert

```
[root@clientB ~]# sealert -l f963dd2e-ab81-4c3a-99ad-4d4d1e6736d5
```

```
SELinux is preventing /usr/sbin/sshd from search access on the directory /opt/userB.
```

```
***** Plugin restorecon (82.4 confidence) suggests *****
```

```
If you want to fix the label.
```

```
/opt/userB default label should be usr_t.
```

```
Then you can run restorecon.
```

```
Do
```

```
# /sbin/restorecon -v /opt/userB
```

```
...
```

```
[root@clientB ~]#
```

SELinux: Caso de la vida real

```
[root@clientB ~]# restorecon -v /opt/userB
restorecon reset /opt/userB context
system_u:object_r:file_t:s0->system_u:object_r:usr_t:s0
[root@clientB ~]#
[root@clientB ~]# ls -lZd /opt/userB/
drwx-----. userB userB system_u:object_r:usr_t:s0 /opt/userB/
[root@clientB ~]#
[userA@serverA ~]$ ssh userB@serverB hostname
clientB.example-rh.com
[userA@serverA ~]$
```

Preguntas?



Más información

- **SELinux Guide**

https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/7-Beta/html/SELinux_Users_and_Administrators_Guide/index.html

- **Fedora Project SELinux Docs**

<http://fedoraproject.org/wiki/SELinux>

- **fedora-selinux-list (mailing list)**

<https://www.redhat.com/mailman/listinfo>

- <http://access.redhat.com> tiene bastantes videos acerca de SELinux. Thomas Cameron, Dave Egts y Dan Walsh han expuesto desde confinamiento de usuarios hasta sandboxing (mls)

- Dan Walsh's blog:

- <http://danwalsh.livejournal.com>

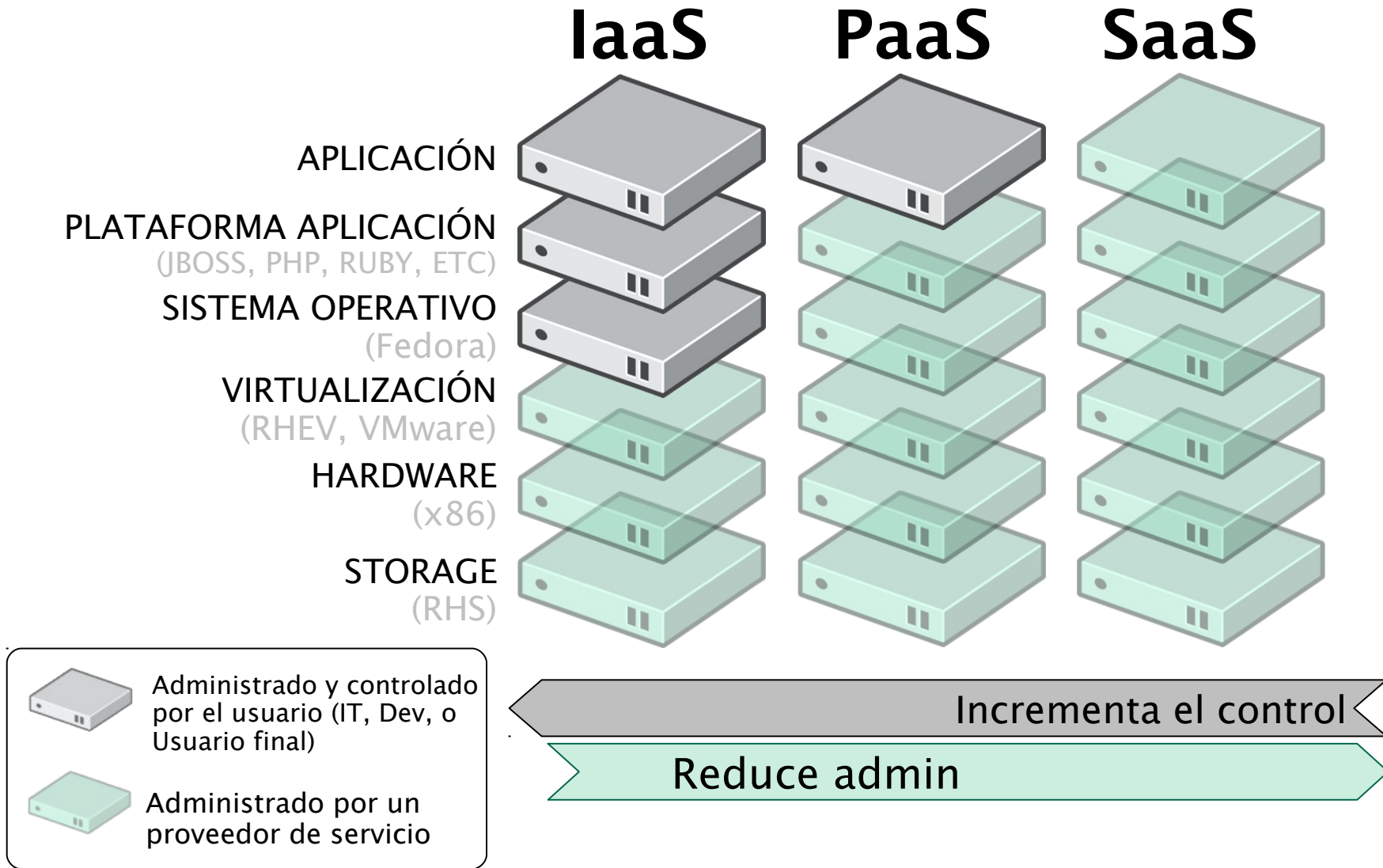
espera, no te hagas guaje

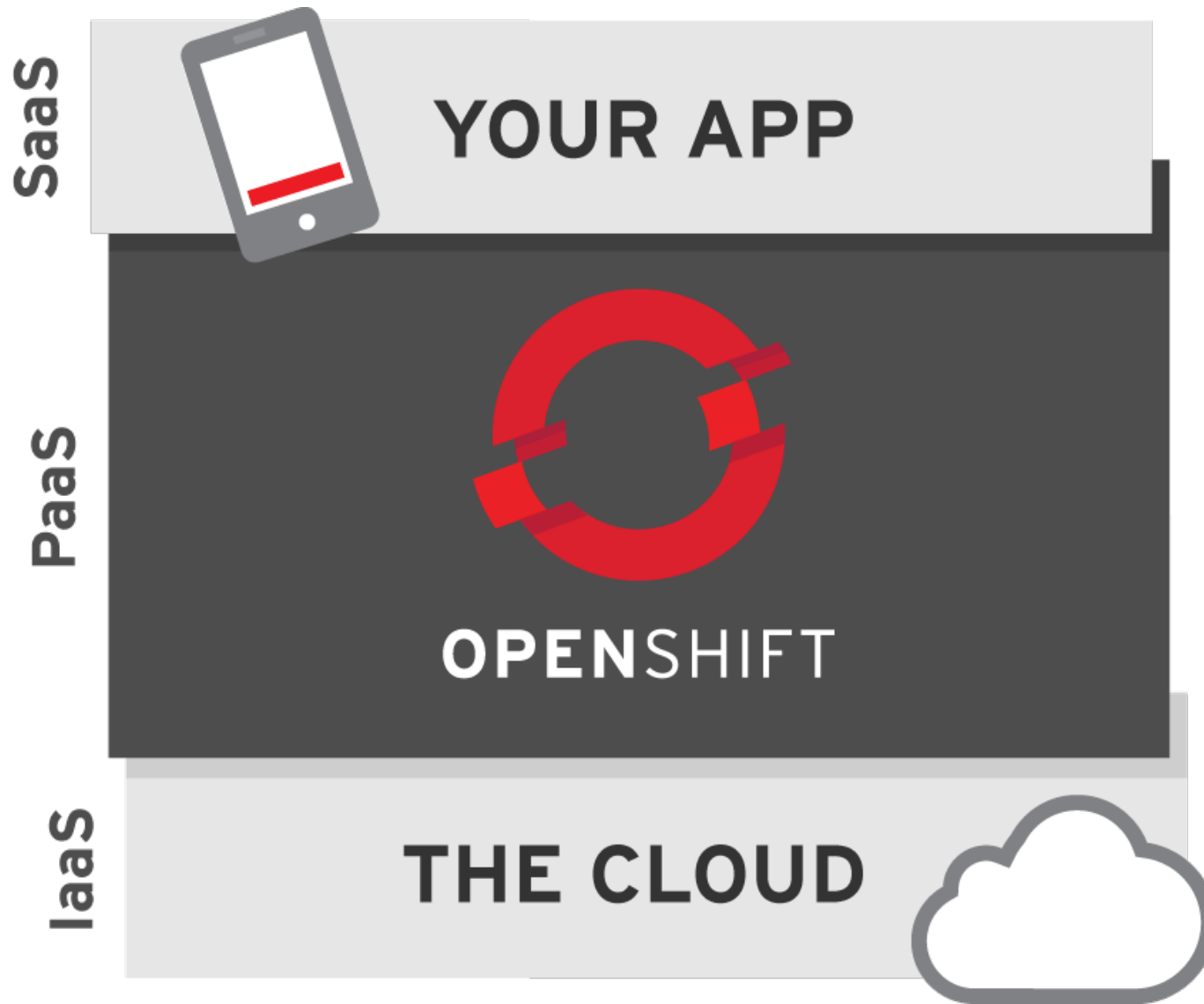
y eso de bloquear a root?

Imagen creada en GeneradorMemes.com



Brevario Cultural: Modelo de Servicios de nube





<http://www.openshift.org/>

Confinando a root

```
root@server:~ 107x25
[root@server ~]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@server ~]# id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@server ~]#
[root@server ~]# runcon -r system_r -t openshift_t -l s0:c0,c1 /bin/sh
sh-4.3#
sh-4.3# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:system_r:openshift_t:s0:c0,c1
sh-4.3# id -Z
unconfined_u:system_r:openshift_t:s0:c0,c1
sh-4.3#
sh-4.3# cat /etc/shadow
cat: /etc/shadow: Permission denied
sh-4.3#
sh-4.3# touch /virus
touch: cannot touch '/virus': Permission denied
sh-4.3#
sh-4.3# ls /
bin  dev  export  lib  lost+found  mnt  proc  run  srv  tmp  var
boot  etc  home  lib64  media  opt  root  sbin  sys  usr
sh-4.3# █
```



Gracias

