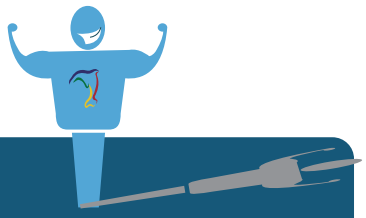
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

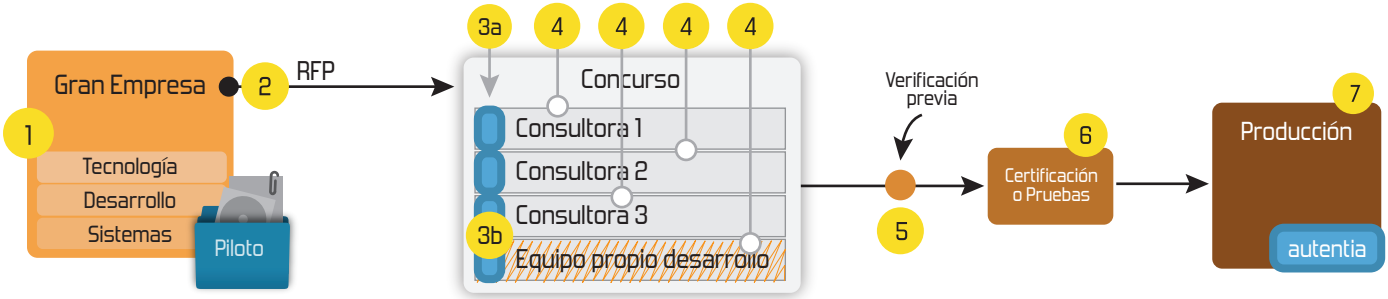
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conociemiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Daniel Ventas López

Becario en autentia.

Ingeniero técnico en informática, especialidad en sistemas.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Ver todos los tutoriales del autor

Fecha de publicación del tutorial: 2013-12-30

Configuración básica de seguridad en servidor linux con iptables y fail2ban

Índice de contenidos

1. Introducción

En informática no todo es desarrollar, también hay muchas otras cosas pendientes como pueden ser securizar los servidores. Para ello vamos a empezar una serie de tutoriales para ver los conceptos básicos para la seguridad en un servidor. Todas las herramientas que expliquemos serán opensource para que se puedan seguir todos los tutoriales.

Por supuesto ninguna herramienta puede garantizar la seguridad de una organización en la que las políticas de seguridad y actualización no estén bien desarrolladas, todavía no se ha inventado el software que nos respalde de eso :), de políticas no se hablará en el tutorial ya que es un tema en sí mismo demasiado extenso.

El campo de la seguridad es tan amplio como se quiera hacer, y cada día se descubren nuevas formas de vulnerar la más sofisticada seguridad, por lo que creo que ningún sistema puede llegar a ser nunca 100% seguro, pero vamos a ponerselo un poco más difícil.

Lo más importante a la hora de empezar con la seguridad, es saber qué es lo que tenemos que proteger y qué es lo que tenemos que dejar público.

En nuestro laboratorio de pruebas vamos a tener un servidor ubuntu, con Apache Tomcat7(acceso público) y acceso por ssh(acceso privado).

Sé que hay mil y una manera de restringir el acceso, en este tutorial vamos a mostrar una de ellas, a través de las iptables.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 17' (2,93 GHz Intel Core 2 Duo, 8GB 1067 Mhz DDR3, 320GB Flash Storage).
- VirtualBox-4.3.6
- Sistema Operativo: Debian wheezy 7.3

3. Iptables

No podríamos pensar en una instalación sencilla de seguridad sin tener en cuenta el firewall, y en este caso vamos a utilizar una herramienta que viene por defecto en la distribución, iptables.

Iptables es una herramienta que permite filtrar, redireccionar, rechazar, encolar paquetes según unas características como pueden ser ip origen o destino, puerto, mac, y un largo etc..

Vamos a definir unos conceptos básicos para comprender cómo funciona la herramienta.

Iptables es una herramienta compleja y tremendamente configurable, con lo que en este tutorial solo vamos a ver una pequeña parte de lo que podemos hacer.

Iptables nos permite definir reglas de distintos tipos como pueden ser de filtrado (por defecto), de nat (para la intranet), de mangle(para manipular paquetes) y de raw(para exceptiones). Cada una de estos tipos tiene una tabla asociada. Para nuestro tutorial sencillo solo vamos a tener en cuenta las reglas de filtrado, que además es la tabla por defecto.

Cada tabla tiene distintas opciones:

- Para la tabla filtros:
  - **INPUT**: Indica paquetes recibidos.
  - **OUTPUT**: Indica paquetes salientes.
  - **FORWARD**: Indica paquetes que se reciben pero que no son para nosotros sino que se enroutan de nuevo.
- Para la tabla NAT:
  - **PREROUTING**: Permite traducir direcciones de paquetes entrantes a ips NAT.
  - **POSTROUTING**: Permite traducir direcciones NAT de paquetes salientes a la correspondiente ip.
  - **OUTPUT**: Indica paquetes salientes.
- Para la tabla Mangle:
  - **INPUT**: Indica paquetes recibidos.
  - **PREROUTING**: Permite traducir direcciones de paquetes entrantes a ips NAT.
  - **POSTROUTING**: Permite traducir direcciones NAT de paquetes salientes a la correspondiente ip.
  - **OUTPUT**: Indica paquetes salientes.
  - **FORWARD**: Indica paquetes que se reciben pero que no son para nosotros sino que se enroutan de nuevo.
- Para la tabla RAW:
  - **PREROUTING**: Permite traducir direcciones de paquetes entrantes a ips NAT.
  - **OUTPUT**: Indica paquetes salientes.

Las reglas se definen mediante target(objetivo) y un criterio. Si se encuentra dentro del criterio se ejecuta el objetivo, si no se pasa a la siguiente regla.

Los objetivos son los siguientes:

- **ACCEPT**: El firewall aceptará el paquete.
- **DROP**: El firewall rechazará el paquete.
- **QUEUE**: El firewall pasará el paquete al espacio de usuario.
- **RETURN**: El firewall dejará de ejecutar el siguiente conjunto de reglas y devuelve el control a la chain llamada.

Para construir reglas iptables solo hace falta saber su sintaxis, ya que vienen incluidas en el núcleo de linux (Viene Netfilter que es el framework para los filtros).

La sintaxis de iptables completa se puede llevar más de un tutorial ella solita, por lo que voy a explicar un pequeño ejemplo para que os hagáis una pequeña idea de cómo funciona y qué es lo que se puede hacer con esta herramienta. Queda en vuestras manos conseguir la configuración deseada en vuestro servidor.

Ejemplo práctico:

Vamos a definir tres reglas:

- Preparar las iptables (se explica abajo).
- Se podrá acceder al puerto 80 (donde está escuchando nuestro Tomcat) desde cualquier dirección, sin restricciones.
- Se podrá acceder por ssh desde cualquier dirección siempre que no se reintente entrar más de tres veces, sino será baneado durante 10 minutos.

Voy a explicar a qué me refiero con preparar las iptables. No es más que poner la máxima restricción, que se rechacen todos los paquetes y luego "abriremos la puerta" solo a los que nos interesen. Para hacerlo es:

```
# IPTABLES -P INPUT   DROP
# IPTABLES -P OUTPUT  DROP
# IPTABLES -P FORWARD DROP
```

Ahora ya podemos ponernos con las reglas "útiles", así que vamos a por la segunda regla que su sintaxis sería la siguiente:

```
 # iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
 # iptables -A OUTPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Como se puede ver, se llama al comando iptables, con -A para indicar la tabla, -p el protocolo, -d/sport el puerto, -m state -- state NEW, ESTABLISHED indica que se aceptarán nuevas conexiones y las ya establecidas. Con -j se indica el objetivo que en este caso es aceptarlas.

Ahora vamos a ver cómo se restringiría el número de intentos por el puerto ssh, quedando baneado durante 10 minutos si se excede del límite.

```
 # iptables -A INPUT -i ${IFaz} -p tcp --dport 22 -m state --state NEW -m recent --set --name SSH
 #Â iptables -A INPUT -i ${IFaz} -p tcp --dport 22 -m state --state NEW -m recent --update --seconds 600 --hitcount 3 --name SSH -j
```

Lo que indicamos con estas dos reglas es que cuente las nuevas conexiones al puerto 22, y cuando superen 3(hitcount), sean baneadas(DROP) 600 segundos.

Podría estar poniendo ejemplos infinitamente porque esta herramienta es muy configurable y seguro que para cada caso tiene una forma de hacerlo, así que os toca investigar a partir de la introducción cómo adaptarlo a lo que necesitáis.

4. Fail2Ban

Siempre hay herramientas que nos ahorran trabajo, y para este caso, además de solucionarnos aspectos básicos en iptables aporta más cosas como revisión de logs. Por eso he elegido ésta herramienta para mostrar cómo podríamos solucionar el tema de las iptables con esta herramienta.

Lo primero que haremos es descargarla, que en Debian es:

```
# apt-get install fail2ban
```

Una vez instalada tenemos que configurarla con nuestras preferencias, para eso tenemos que editar el archivo /etc/fail2ban/jail.conf, en el cual están las "jaulas" de las herramientas que queramos securizar.

Primero un repaso, esta herramienta utiliza los logs y los compara con una serie de filtros que vienen en la instalación y están en la carpeta /etc/fail2ban/filter.d/, y en cada jaula que activemos podremos marcar qué filtro queremos activar, así como otras características como dónde recoge los logs, el puerto, etc.. En nuestro caso supondremos la instalación por defecto, por lo que no nos hará falta cambiar nada de la configuración, solo activar los filtros a nuestras preferencias.

Empezamos configurando las ip que debe ignorar para los filtros, en nuestro caso agregaremos la red local, y el tiempo de baneado, que 10 minutos me parecen poco así que se pondrá una hora (el tiempo se marca en minutos). Se dejará comentado la línera original para comprobar el cambio.

```
ignoreip = 127.0.0.1/8 192.168.1.0/24
#ignoreip = 127.0.0.1/8
bantime  = 3600
#bantime = 600
maxretry = 3
```

A continuación marcamos el email de destino para los avisos y configuramos el nivel de aviso, en nuestro caso lo dejamos por defecto.

```
destemail = admin@system.com
#destemail = root@localhost

...

action = %(action_)s
```

En action podemos elegir entre los tres tipos de avisos que vienen por defecto:

- **action_**: Es la que menos información manda, solo manda información de baneados.
- **action_mw**: Además de banear avisa por email de quién ha baneado.
- **action_mwl**: Es la más verbosa y además de la última envía al email líneas relevantes del log como cuándo se inicia y se detiene el servicio, etc..

Ahora ya pasamos a activar nuestras jaulas, como tenemos todo por defecto, solo nos hace falta cambiar el atributo "enabled = false" por "enabled = true" en las herramientas que queramos controlar. En este caso, como tenemos el mismo laboratorio de pruebas, vamos a activar la opción apache y ssh.

```
[ssh]

enabled  = true
#enabled = false
port     = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 6

[apache]

enabled  = true
#enabled = false
port     = http,https
filter   = apache-auth
logpath  = /var/log/apache*/*error.log
maxretry = 6
```

Como véis es muy fácil de configurar, y ya sólo nos queda reiniciar el servicio con:

```
# service fail2ban restart
```

Y podemos comprobar que han cambiado las iptables con el comando:

```
# iptables -L

Chain INPUT (policy ACCEPT)
target     prot opt source               destination
fail2ban-apache  tcp  --  anywhere             anywhere             multiport dports http,https
fail2ban-ssh  tcp  --  anywhere            anywhere             multiport dports ssh
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain fail2ban-apache (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere

Chain fail2ban-ssh (1 references)
target     prot opt source               destination
RETURN     all  --  anywhere             anywhere
```

Como vemos se ha creado una tabla para cada servicio que le hemos indicado, que será donde irá añadiendo las reglas que vayan haciendo falta.

Para ver el estado del servicio es:

```
 # fail2ban-client status

 Status
|- Number of jail:  2
`- Jail list:       apache, ssh
```

5. Conclusiones

Con esto creo que es suficiente como introducción a las iptables, espero que os ayude en los inicios de configuración de linux y recordad que en seguridad nada es sificiente pero tampoco vamos a ponerselo fácil :).

Un saludo y cualquier duda en los comentarios.