Hardening del servidor

¿Qué es?

En este apartado vamos a securizar todavía más el servidor. Esto lo hacemos junto a la securización o endurecimiento conseguido a través del firewall (iptables) y la configuración lógica, es decir, la creación de usuarios y grupos necesarios para cumplir con el requisito de **mínimo privilegio**.

Configurar apache

Práctica 1 Realiza y documenta este punto y el siguiente. Como resultado, debes crear un docker con una imagen de apache configurada como aquí se indica

Para cumplir con el requisito de **mínima exposición** vamos a eliminar todos aquellos módulos que no nos vayamos utilizar.

Para conocer qué módulos están activos (aquellos que están en la carpeta `mods_enabled`). Podemos hacer un listado mediante el siguiente comando

```
sudo apache2ctl -t -D DUMP_MODULES
```

o mediante

```
kali@kali:~$ sudo a2dismod
Your choices are: access_compat alias auth_basic authn_core authn_file authz_core authz_host
Which module(s) do you want to disable (wildcards ok)?
```

Por ejemplo vamos a eliminar el módulo `mod_autoindex`. Si no sabes qué función tiene este módulo, puedes visitar esta página de apache. En breves palabras, impide la creación automática de una página `index` cuando no encuentra el archivo (`index.html` o `index.php`) en la carpeta que se visita. Esta página generada automáticamente muestra todos aquellos archivos y directorios de la carpeta en cuestión.

Otra opción que debemos suprimir es que **apache** no devuelva el tipo de servidor (o al menos la versión). De esta forma no damos pistas al atacante de la versión y/o tipo de servidor.

Por ejemplo, si lanzamos el comando:

```
curl --head localhost
```

que devuelve

```
HTTP/1.1 200 OK
Date: Mon, 15 Mar 2021 08:00:08 GMT
Server: Apache/2.4.46 (Debian)
Content-Type: text/html; charset=UTF-8
```

También se puede comprobar en las cabeceras de respuesta mediante el navegador

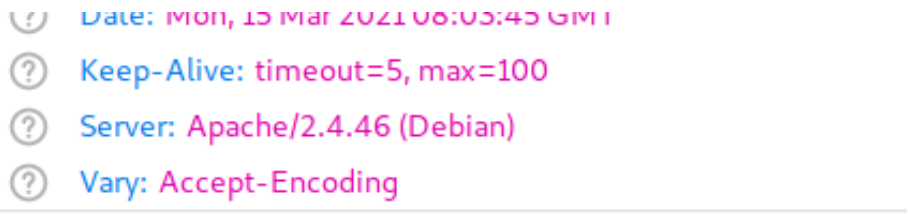

Figure 1: image-20210315090439867

Para evitar que muestre la signatura debemos modificar la configuración de apache en el archivo `/etc/apache2/apache2.conf` y añadir `ServerTokens ProductOnly`, de esta forma sólo pueden consultar la versión los módulos de apache `ServerSignature Off`, eliminamos completamente la signatura

Web Application Firewall (WAF)

> Según la Wikipedia
>
> Un firewall de aplicaciones web (WAF) es un tipo de firewall que supervisa, filtra o bloquea el tráfico HTTP hacia y desde una aplicación web. Se diferencia de un firewall normal en que puede filtrar el contenido de aplicaciones web específicas, mientras que un firewall de red protege el tráfico entre los servidores. Al inspeccionar el tráfico HTTP un WAF protege a las aplicaciones web contra ataques como los de inyección SQL, XSS y falsificación de petición de sitios cruzados (CSRF).

En 2002 se creó el proyecto de código abierto ModSecurity para hacer la tecnología WAF más accesible y resolver los obstáculos dentro de la industria, como casos de negocios, barreras de costos y los conjuntos de reglas particulares de cada empresa. ModSecurity creó un conjunto de reglas básicas para proteger las

aplicaciones web, basado en las vulnerabilidades detectadas por el OASIS Web Application Security Technical Committee's (WAS TC). En 2003, este trabajo fue ampliado y estandarizado con la creación de la Lista Top 10 del Open Web Application Security Project's (OWASP). OWASP publica con cierta regularidad una lista con los 10 riesgos de seguridad más críticos de las aplicaciones web. Esta lista se convertiría en la referencia de la industria para muchos temas de seguridad en la web.

Configurarlo es bastante complicado ya que funciona por reglas por las que aceptamos o rechazamos peticiones. Pero la OWASP provee una configuración por defecto que incluye una protección para las reglas más comunes. Así que lo mejor es empezar por este conjunto de reglas y luego ir añadiendo las propias.

Una solución de compromiso para no dar todas las reglas, se muestra una configuración que tiene OWASP. Para instalarlo en nuestro servidor:

https://coreruleset.org/installation/

https://resources.infosecinstitute.com/topic/configuring-modsecurity-firewall-owasp-rules/

1. Hacemos un clon del repositorio
2. Instalar `libapache2-mod-security2`
3. Configurarlo:

- Ir a https://github.com/coreruleset/coreruleset/blob/v3.4/dev/INSTALL
- ir a la carpeta de las reglas ir a `/etc/modsecurity` y hay que modificar el .conf y poner SecRuleEngine a On
- Luego ir a /etc/apache2/mods-available y hacer un cat de security2.conf
- Se reinicia apache y ya va

apache extra

Si queremos que sólo sirva tráfico a una IP, en 000-default.conf

```
<Location />

Require ip 192.168.1.24

</Location>
```

De esta forma sólo le damos acceso a nuestro reverse proxy

Otra configuración por defecto que es recomendable el .htaccess. Se debe prohibir el uso de este archivo ya que es una gasto de recursos al hacerlo en caliente.

MySql

Restringir sólo el puerto local y nunca a la máquina remota

Dentro de `/etc/mysql` está el archivo `my.cnf`

dentro de `[mysqld]` poner

```
bind-address = 127.0.0.1
```

Reiniciar y ya sólo deja acceder desde localhost

Otra configuración interesante es eliminar la posibilidad de que mysql pueda leer cualquier archivo del sistema y eso es muy inseguro. Esta inclusión se realiza mediante load_file('/etc/apt/sources.list')

Para eliminar este permiso para listar archivos.

en mariadb.conf.d en [mysqld]

```
local-infile = 0
```

```
secure-file-priv = /dev/null
```

Otra medida es renombrar el usuario root. Para ello,

```
update mysql.user set user="ciberseguridad" where user="root"
flush privileges
```

Privilegios de los usuarios: De forma homologa a los que ocurre en el sistema linux, en MySQL debemos tener una correcta gestión de los usuarios, ya sean para personas o cuentas de servicio para dar acceso a las aplicaciones, otorgando solo los permisos necesarios de los datos necesarios para cumplir con el requisito de mínimo privilegio