Seguridad

Muy genérico pero aquí voy a poner tips que nos facilitan la vida para implementar procedimientos que permiten aumentar nuestra seguridad

- Host dinámicos
- Limitación de acceso por host dinámico - Apache y Nginx
- Llaves SSH en entornos *nix (linux, MacOS X, Windows WSL)
- Deshabilitar 2FA en WHMCS para un administrador
- Error de acceso SSH con llaves ssh (Permission denied (publickey)) habiendo verificado que las llaves son correctas

Host dinámicos

Introducción

Un **host dinámico** es una entrada FQDN con un componente dinámico en la asignación de su IP.

Es un mecanismo muy apropiado para poder añadir una capa extra de seguridad a los accesos a determinados sistemas y programas.

> Un fallo común en la aplicación de políticas de acceso a dispositivos o sistemas protegidos mediante cortafuegos (firewall) es el añadido de IP tratadas como fijas que no lo son, o fijas que en un momento determinado cambian o deberían dejar de estar autorizadas.

Requisitos

- Una cuenta en un proveedor de DNS dinámico (aquí, No-IP) y un correo para confirmar el alta.
- El software cliente de No-IP en el equipo cuya IP publicar (Windows, Mac, Linux o móvil).
- El cliente DNS dinámico aprobado por el firewall o servicio que va a autorizar por hostname (p. ej. la lista dyndns de CSF).

Aplicaciones de un host dinámico

Son muchas y variadas siendo las más importantes para el escenario privado o empresarial del hosting las siguientes:

- Limitar el acceso a servidores o dispositivos vía SSH mediante host autorizado
- Limitar el acceso a páginas web o zonas de una página web via host autorizado

¿Cómo obtener y mantener un host dinámico?

Para tener un host dinámico necesitamos un sistema de DNS que nos permita mediante una aplicación actualizar regularmente (300 segundos +/-) los datos de nuestro host dinámico.

Hay algunos de pago y otros gratuitos pero yo me decanto por noip de DNS Services.

Proceso de alta en Noip.com

El alta como usuario es gratis, y nos permitirá hasta 3 hostnames de un único dominio.

Alta en NoIp

Durante el proceso deberemos escoger el nombre de host y el dominio del que derivará nuestro hostname dinámico.

Una vez aceptado, deberemos confirmar nuestra dirección de correo electrónico.

Acceso a noip.com

> Importante: Noip no permite que tengas instaladas las Vue Developers Tools (todavía no entiendo por qué)

Deberemos completar por seguridad los datos personales y de uso.

Descargamos el software para que podamos actualizar nuestro ordenador, tablet o móvil.

Instalamos y configuramos.

Configuración del software de noip

Instalación

Instalación NoIp Software

Primer acceso

En la primera apertura veremos que falta por indicar el host dinámico que vamos aplicar a esa instalación. Seleccionamos y listo.

Primer acceso

Configuración

Configuración

Funcionando

Funcionando no ip

Consulta su ayuda para ver opciones (arranque automático, etc).

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Verificación

- `nslookup <tu-host>.ddns.net` (o el dominio elegido) debe resolver a la IP pública actual del equipo; al cambiar la IP, el cliente de No-IP debe actualizarla en unos minutos.
- En el propio cliente, la captura "Funcionando" es el estado esperado: hostname seleccionado y estado activo.
- La prueba final es el consumo: el firewall o servicio que autoriza por hostname debe dejar pasar el acceso desde ese equipo, y solo desde él.

Limitación de acceso por host dinámico - Apache y Nginx

Introducción

Muchas veces necesitamos usar mecanismos de seguridad para el acceso a determinadas áreas, más cuando nos encontramos con software que ya ha tenido más de una incidencia de seguridad o conocemos la baja calidad de su software como WHMCS.

Uno de los mejores mecanismos es la aplicación de seguridad mediante el acceso a la zona que queremos proteger, mediante el uso de Host dinámicos

Requisitos

- Un host dinámico operativo (ver la página enlazada) o una IP fija autorizada.
- Apache 2.4 con `mod_authz_host` (para la vía Apache); para Nginx, el módulo rdns compilado e instalado (ver su sección).

Apache

Con Apache usaremos *.htaccess* para cerrar todas las conexiones a la zona web que queremos proteger.

Requerimientos

- authz_host_module, generalmente forma parte del binario de nuestro sistema o está cargado como módulo. Apache Module mod_authz_host

Implementación

```
Require forward-dns mi.hostdinamico.tld
```

También si tenemos IP fija podemos usarla (aunque en nuestro tip anterior indico mi preferencia por los host dinámicos). Advertencia: tal como quedó escrito, el bloque usa `Require forward-dns` con una IP; para IPs la directiva correcta de mod_authz_host es `Require ip`:

```
Require forward-dns 00.00.00.00
```

Nginx

** pendiente de testing **

Nginx tiene otros mecanismos, y lo describe en HTTP rDNS

Requerimientos

- módulo ngx_http_rdns_module

Advertencia: el bloque siguiente quedó mal cerrado en el original (el `location` final vacío y `echo_exec` sin contexto): trátalo como esqueleto, no como configuración operativa, y complétalo según la documentación del módulo antes de usarlo.

```
server {
resolver 127.0.0.1; # Si tienes activado dns localhost, sino pon tu resolver preferido

    rdns_allow mi\.hostname\.tld;

    location /path {
    echo_exec $protegido
    }

    location
}
```

Compilación

En mi blog tengo una entrada sobre LEMP en la que hago compilación de módulos como este.

Para compilar este módulo de forma dinámica:

```
$ cd ~/soft/
$ git clone git@github.com:flant/nginx-http-rdns.git
## $version ya estaba declarada en la compilacion sino, poner la que corresponda. Ten en
cuenta que este post
## entiende que se esta siguiente el modelo del enalce de más arriba
$ cd ninx-$version/
$ /configure --with-compat --add-dynamic-module=../nginx-http-rdns
$ make
$ make modules
$ sudo mv objs/ngx_http_rdns_module.so /usr/lib/nginx/modules/
```

Añadir la carga del módulo a nginx en /etc/nginx/modules-enabled/51-mod-http_rdns.conf

```
load_module modules/ngx_http_rdns_module.so;
```

Configurar en los sitios que se requiera según su configuración.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Verificación

- Apache: desde el host dinámico autorizado la zona debe cargar (200); desde cualquier otra red debe denegarse (403).
- Nginx: `sudo nginx -t` debe validar la configuración con el módulo cargado antes de recargar, y la zona protegida debe responder igual: autorizada desde el host dinámico, denegada desde fuera.
- Tras cada cambio de IP del host dinámico, repetir la prueba autorizada: es la comprobación de que la resolución rDNS sigue siguiendo a la IP.

Llaves SSH en entornos *nix (linux, MacOS X, Windows WSL)

Introducción a las llaves SSH

El uso de llaves SSH es primordial para operaciones con servidores, ya sean aisladas o automatizadas. Añaden una capa extra de seguridad frente al uso de contraseñas y una versatilidad a las operaciones que se hace necesario tener nuestro par de llaves SSH.

Podemos tener nuestro par de llaves único (acosenjado por cuestiones de seguridad, ya que su revocación es más fácil) o por dispositivo.

Llaves OpenSssh (OpenSssh Keys)

Como norma general los servidores unix usan OpenSSh Server y por tanto sus llaves requieren que esten en este formato. Si usamos un entorno tipo unix (Linux, MacOS X, Windows WSL, FreeBsd) ese formato no supondra ningun problema. Si usamos Windows sin WSL, entonces seguramente usemos algun programa como Putty que tiene su formato cerrado, y el procedimiento requiere convertir sus llaves publicas a formato OpenSSH.

> “ Puedes consultar nuestro manual Manual para conexión SSH con Putty

Generar un par de llaves

Vamos generar un par de llaves OpenSSH, sin contraseña, para operacioneas desatendidas y porque creemos que es suficiente el nivel de seguridad que no sofrece, sin necesidad de añadir la capa extra con la contraseña.

Tan sencillo como posicionarnos en nustro home, y ejecutar el comando, admitiendo todos los pasos por defecto.

Eso creara un par de llavez con el nombre `id_rsa` (privada) y `ìd_rsa.pub` (publica)

```
yamna@nox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/yamna/.ssh/id_rsa):
Created directory '/home/yamna/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/yamna/.ssh/id_rsa
Your public key has been saved in /home/yamna/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:20RVkgzXUBA9/V2tkoDejoim+L7zAtjvfxvva4CmQaA yamna@nox
The key's randomart image is:
+---[RSA 3072]----+
|         ..oBXo..|
|  .     . .oo.+ +|
| . .   . ... . o+|
|E   .   ... o . o|
|.. . . oSo.  .   |
|o . + + o+.      |
| o + +  o..      |
|. + o   .o.      |
| o+*o....++.     |
+----[SHA256]-----+
```

ssh-keygen

Parte pública

Si queremos acceder a un servidor remoto (root o usuario) deberemos añadir la llave publica a el fichero de autorizaciones del sistema remoto o bien facilitarselo al administrador para que el haga su trabajo.

No existe ningun problema en enviar esta llave (la pública) a alguien, incluso en modo texto. A fin de cuentas es la llave pública que no sirve nada más que para que otros la ulticen para **autorizarnos** a nosotros.

Parte privada

Esta es la parte que si tenemos que atender a su seguridad. La perdida de un dispositivo, que tenga esa llave, unido a que el hacker pueda analizar nuestro historico, le dará la oportunidad de acceder de forma inmediata a todos los servicios en los que tu par de llaves esten autorizadas. Asi pues, la más minima sospecha de intrusión, substracción, perdida de dispositivo, con ese par de llaves, conlleva a la revocación inmediata de las llaves y aviso a los administradores implicados. (como si perdieras la contraseña)

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como esta, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Deshabilitar 2FA en WHMCS para un administrador

Introducción

En muchos entornos de trabajo es posible que lidiemos con los problemas de pérdida o ausencias técnicas de los administradores que están obligados al uso de la verificación de contraseña **2FA** o doble factor.

Además, a veces es una labor muy difícil la de formar al personal de las empresas en sus obligaciones de seguridad, por lo que muchas veces pierden o no tienen acceso (porque no lo descargaron, no lo guardaron...) al o los **códigos de seguridad**.

Requisitos

- WHMCS con acceso a la base de datos (SSH con cliente mysql o phpMyAdmin) y credenciales de la BD.
- El nombre de usuario del administrador afectado.
- Consciencia de seguridad: esto desactiva el 2FA de ese administrador; tras el acceso, volver a activarlo y guardar los códigos de respaldo es parte del proceso, no un extra.

Desactivación de la obligación de uso de 2FA para un administrador de WHMCS

Debemos acceder vía SSH o phpMyAdmin y ejecutar dos updates en la base de datos de nuestro WHMCS:

```
# $ mysql # Podemos tener acceso automatizado o necesitar escribir las credenciales
#
$ mysql -u user_whcms -p
mysql > update whmcs_bd.tbladmins set authmodule='' where username='nombre_de_usuario_admin';
mysql > update whmcs_bd.tbladmins set authdata='' where username='nombre_de_usuario_admin';
```

Los dos updates limpian el módulo de autenticación (`authmodule`) y sus datos (`authdata`) para ese administrador: sin módulo, WHMCS no exige el segundo factor en el siguiente login.

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido se entrega, tal y como está, sin que ello implique ninguna obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Verificación

- `SELECT username, authmodule, authdata FROM tbladmins WHERE username='nombre_de_usuario_admin';` debe devolver ambos campos vacíos tras los updates.
- El login del administrador en el panel de WHMCS debe completar solo con usuario y contraseña, sin pedir el código 2FA.
- Post-acceso: reactivar el 2FA del administrador y archivar los códigos de respaldo (el escenario que motivó el problema no debe repetirse).

Error de acceso SSH con llaves ssh (Permission denied (publickey)) habiendo verificado que las llaves son correctas

Introducción

Desde OpenSSH 7.0 se deshabilito por defecto el intercambio de claves SSH-DSS. Esto puede producir un comportamiento de error, muy silencioso que es fácil de detectar y corregir.

Si tenemos una llave creada con el tipo dss y esta está autorizada en un sistema, o por error, no vimos que nos pasaban una llave con esa característica, para añadirla a un sistema, esta no funcionará en el sistema actualizado, y nos dará en primera instancia un error de **Denegación por llave incorrecta**.

Análisis

En el acceso general con esa llave sólo recibiremos un mensaje corto

```
root@server:~# ssh  user_remoto@servidor.remoto.tld -p 51514
user_remoto@servidor.remoto.tld: Permission denied (publickey).
```

Como siempre, debemos acostumbrarnos a analizar nuestros problemas con el aumento del sistemas de logs o de mensajes (debug)

Dejaremos el listado sólo en lo justo para ver el problema

```
root@server:~#  ssh  user_remoto@servidor.remoto.tld -p 51514 -vv
OpenSSH_8.2p1 Ubuntu-4ubuntu0.3, OpenSSL 1.1.1f  31 Mar 2020
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files
debug1: /etc/ssh/ssh_config line 21: Applying options for *
...
debug1: Trying private key: /root/.ssh/id_rsa
debug1: Trying private key: /root/.ssh/id_ecdsa
debug1: Trying private key: /root/.ssh/id_ecdsa_sk
debug1: Trying private key: /root/.ssh/id_ed25519
debug1: Trying private key: /root/.ssh/id_ed25519_sk
debug1: Trying private key: /root/.ssh/id_xmss
debug2: we did not send a packet, disable method
debug1: No more authentication methods to try.
```

Importante que nos indique que ha buscado la llave y no le ha valido.

Revisando la llave vemos que tiene el tipo ssh-dss el cual no está aceptado por el servidor remoto.

```
También es importante ver el log del **servidor sshd** si es posible, y dependiendo del nivel de log que tenga podremos ver que nos indica que el tipo dss no está aceptado
```

Revisamos el valor de la llave y vemos en el incio, que tenemos una llave tipo DSS.

```
root@server:~# cat .ssh/id_dsa.pub
ssh-dss AAAAB3N….
```

Opciones

Reconstrucción de la llave con formato rsa (Recomendado)

El tipo RSA es el más aconsejado. Si el sistema no te crea la llave por defecto con DSA o no se la crea al usuario que te está facilitando la llave, deberá forzar el comando de generación de sus llaves para que use el formato RSA.

```
root@server:~#  ssh-keygen -t rsa
```

Deberemos actualizar la llave en nuestro fichero de authorized_keys, y ya debería funcionar

Activación del algoritmo DSS en el servidor (No aconsejable)

Puede existir algún escenario, por prisas, situación específica, etc, que no nos permita de forma inmediata, el recrear la llave del servidor que quiere acceder, por lo que debemos reactivar el tipo de llave que tenemos.

Es imperativo, entender que esto es un atajo, muy común en los artículos de internet, que puede llevar a dejar nuestro sistema más inseguro, si luego no lo devolvemos a la normalidad.

Ese error, ocurre en el 90% de los casos, porque el administrador, va corriendo, y luego no vuelve atrás a corregir la situación.

Para ello debemos editar el fichero `/etc/ssh/sshd_config` añadiendo la línea siguiente:

```
PubkeyAcceptedKeyTypes=+ssh-dss
```

Centos 8

Requiere un paso adicional, relacionado con su System-wide Cryptographic Policies.

Primero deberíamos conocer el nivel actual (suele ser por defecto) para luego devolverlo a su estado original

```
$ update-crypto-policies --show
DEFAULT
```

```
Debemos, devolver en cuanto podamos, (nueva llave) nuestro sistema a la posición de seguridad inicial
```

Otros enlaces

- OpenSSH 7.o Release notes
- Chapter 5. Using system-wide cryptographic policie
- SSH keeps skipping my pubkey and asking for a password

- SSH connection issue on Fedora 30 with OpenSSH 8.0 (Using ssh-dss), still asking for password with public key setup

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.