Seguridad, SSl, etc.

Todos sobre seguridad, certificados, comprobación de ciphers

- Verificar versiones y ciphers soportados de TLS, SSL
- CSF error: *WARING* URLGET set to use LWP but perl module is not installed, fallback to using CURL/WGET
- Limitar en el tiempo (expirar en una fecha) una llave openSSH en el authorized_keys
- Comprobar una conexión SMTP autentificada en el shell con SSL o TLS
- CSF Firewall: añadir IPs al deny de forma definitiva.
- Certificados Letsencrypt sin servidor web o sin resolver en el servidor web

Verificar versiones y ciphers soportados de TLS, SSL

Introducción

Una de los mayores problemas al que se enfrenta el soporte es lidiar con los clientes y los problemas derivados del uso de sistemas (windows, MacOSX, Ios, Android, ...) obsoletos.

Verificar TLS soportado por un protocolo

```
h=nombre_del_host
p=port
## Tls 1.2
openssl s_client -connect $h:$p -tls1_2
### Tls 1.1
openssl s_client -connect $h:$p -tls1_1
### Tls 1
openssl s_client -connect $h:$p -tls1
```

El retorno debe tener una linea como esta

```
Verification: OK
```

Enumerar los ciphers ssl

```
h=nombre_del_host
p=port
nmap --script ssl-enum-ciphers -p $p $h
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-15 17:43 CEST
Nmap scan report for kvm468.ceinor.com (5.135.93.99)
Host is up (0.056s latency).
```

```
PORT    STATE SERVICE
465/tcp open  smtps
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (dh 2048) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (secp256r1) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (secp256r1) - A
|     compressors:
|       NULL
|     cipher preference: client
|_  least strength: A

Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
```

Links y agradecimientos

- Command prompt to check TLS version requiere by a host
- Checking ssl tls Version Support of a Remnote Host from Command line

CSF error: *WARING* URLGET set to use LWP but perl module is not installed, fallback to using CURL/WGET

Introducción

Casi siempre que se instala CSF Firewall en una distribución linux tenemos el mismo problema relativo a la ausencia de las librerías de perl LWP requeridas por CSF o mejor dicho preferidas ya que el propio CSF nos ofrece 3 posibilidades de configuración para la descarga de listas u otros elementos desde sitios remotos vía HTTPS.

1. HTTP::Tiny
2. LWP::UserAgent
3. CURL/WGET (set location at the bottom of csf.conf)

LWP::UserAgent

Es la opción deseada y configurada por CSF, y esta configuración al no encontrar disponible el binario correspondiente, te lanza la advertencia cada vez que ejecutes un comando de csf.

```
*WARNING* URLGET set to use LWP but perl module is not installed, fallback to using CURL/WGET
```

Realmente en su propio fichero de configuración se encuentra el cómo, pero si has llegado hasta aquí, es que no lo leiste. (Es lo normal. No solemos leer los LEAME.txt vamos a leer los cientos de comentarios de algunos ficheros de configuración)

Distribuciones basada en RPM (Centos, AlmaLinux, CloudLinux,..)

```
yum install perl-libwww-perl.noarch perl-LWP-Protocol-https.noarch
```

Distribuciones basadas en APT (Debian, Ubuntu,...)

```
apt-get install libwww-perl liblwp-protocol-https-perl libgd-graph-perl
```

Via cpan

```
# perl -MCPAN -eshell
cpan> install LWP LWP::Protocol::https
```

Documentación original CSF (ConfigServer Firewall)

```
# The following option can be used to select the method csf will use to
# retrieve URL data and files
#
# This can be set to use:
#
# 1. Perl module HTTP::Tiny
# 2. Perl module LWP::UserAgent
# 3. CURL/WGET (set location at the bottom of csf.conf if installed)
#
# HTTP::Tiny is much faster than LWP::UserAgent and is included in the csf
# distribution. LWP::UserAgent may have to be installed manually, but it can
# better support https:// URL's which also needs the LWP::Protocol::https perl
# module
#
# CURL/WGET uses the system binaries if installed but does not always provide
# good feedback when it fails. The script will first look for CURL, if that
# does not exist at the configured location it will then look for WGET
#
# Additionally, 1 or 2 are used and the retrieval fails, then if either CURL or
# WGET are available, an additional attempt will be using CURL/WGET. This is
# useful if the perl distribution has outdated modules that do not support
# modern SSL/TLS implementations
#
# To install the LWP perl modules required:
#
# On rpm based systems:
#
#   yum install perl-libwww-perl.noarch perl-LWP-Protocol-https.noarch
#
```

```
# On APT based systems:
#
#   apt-get install libwww-perl liblwp-protocol-https-perl
#
# Via cpan:
#
#   perl -MCPAN -eshell
#   cpan> install LWP LWP::Protocol::https
#
# We recommend setting this set to "2" or "3" as upgrades to csf will be
# performed over SSL as well as other URLs used when retrieving external data
#
# "1" = HTTP::Tiny
# "2" = LWP::UserAgent
# "3" = CURL/WGET (set location at the bottom of csf.conf)
URLGET = "2"

# If you need csf/lfd to use a proxy, then you can set this option to the URL
# of the proxy. The proxy provided will be used for both HTTP and HTTPS
# connections
URLPROXY = ""
```

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.

Limitar en el tiempo (expirar en una fecha) una llave openSSH en el authorized_keys

Introducción

Un serio handicap en los modelos de administración de sistemas basados en el ser humano, es que un despiste puede ser fatal para nuestra seguridad.

Cuando necesitamos añadir una llave SSH a un servidor para el acceso del propietario de dicha llave, y necesitamos que sea temporal, suele ocurrir, que al final, la llave termina por olvidarse.

> Por mi trabajo, muchas veces accedo a servidor en los que tengo que realizar trabajos de auditoría o de tunning, y me encuentro llaves autorizadas desde el principio de los tiempos.

Añadir un limite de tiempo a una llave OpenSSH autorizada

Al igual que podemos limitar los comandos que el propietario de dicha llave SSH pueda hacer en nuestro sistema, también podemos limitar la validez de la llave en el tiempo.

Si nuestro servidor esta ejecutando una version OpenSSH 7.7 o superior, podremos hacerlo añadiendo `expiry-time` a la entrada de la llave a la que queremos limitar en el tiempo el acceso con el formato `expiry-time="YYYYMMDD" ssh-rsa AAAAB3Nz...w== Algun comentario`

```
expiry-time="20210621" ssh-rsa AAAAB3NzaC1yc2.. ...MXhBut9HKkWI9/ root@prox03
```

También puedes especificar un tiempo más concreto, usando `YYYYMMDDhhmm` (la versión `YYYYMMDD` entiende como si fuera la media noche, `2020-06-21 00:00:00`

Enlaces

Adding expiration date to SSH key

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.

Comprobar una conexión SMTP autentificada en el shell con SSL o TLS

Introducción

Muchas veces el soporte técnico o nosotros, tenemos que comprobar si los datos que tenemos y si el servidor **SMTP remoto**, están operando correctamente. No es necesario hacer como hacen algunos un cambalache creando una cuenta en el programa de correo electrónico, sino que como casi siempre podemos acudir a nuestra shell para realizar las pruebas pertinentes.

Comprobar la autenticación SMTP y la conexión SSL usando la línea de comandos o shell.

Autenticación SMTP

La autentificación (autenticación) es el mecanismo por el cual un usuario se identifica a sí mismo en un **servicio** de un servidor. En este caso el servicio es el **correo electrónico saliente** o **SMTP** y es necesario para que podamos **enviar** correo electrónico.

Preparación, prueba y verificación

Para hacer la prueba es necesario tener instalado el paquete **openssl** de nuestro ordenador.

Crear la cadena de autenticación para una login basado en PLAIN

> Generalmente los servidores de correo electrónico, usan como medio de autenticación uno denominado **PLAIN**, que consiste en pasar un texto plano (ASCII) que contiene el par **usuario + contraseña**

Antes de realizar la prueba debemos obtener la cadena de caracteres ASCII que contiene el par usuario_smtp + contraseña.

Usando Bash

En el momento de escribir esto, el tip que tenía en mi entrada original Cómo comprobar la autenticación SMTP SMTP Auth y la conexión con StartTLS en el shell me da error. La verdad es que he comprobado si había un error en mi escritura, y revisado con otros colegas. Así que he optado por no hacer el comando en una línea sino dividirlo en dos que si me funciona

```
$ echo -ne usuario@servidor.smtp.com | base64
emFiYml4QQDlbnRyYWwuY2FzdHJpcy5jb20=
$ echo -ne 4Mmr8Hop3FsmQvKtb8Ei | base64
NE1tcjVUb3BhRnNtUXZLdGI4RWk=
```

Ahora conectamos vía openssl

> h y p son variables de entorno para poder trabajar más fácilmente

El puerto deberá ser el apropiado a la conexión, en este caso **startssl**

```
$ h=servidor.smtp.com
$ p=455
$ openssl s_client -connect $h:$p -starttls smtp
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = R3
verify return:1
depth=0 CN = servidor.smtp.com
verify return:1
---
Certificate chain
 0 s:CN = servidor.smtp.com
...
```

```
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

Esto ya nos indica que el servidor está activo, escuchando en el puerto solicitado, y admitiendo la conexión vía startssl

Ahora podemos usar `EHLO there` para obtener los comandos disponibles

```
EHLO there
250-servidor.smtp.com
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
```

O pasar directamente a la autenticación

```
AUTH LOGIN
334 VXNlcm5hbWU6
emFiYml4QQDlbnRyYWwuY2FzdHJpcy5jb20=
334 UGFzc3dvcmQ6
NE1tcjVUb3BhRnNtUXZLdGI4RWk=
235 2.7.0 Authentication successful
```

Usando Perl

Si el usuario contiene la @ esta deberá escaparse con la barra invertida (\) de otra manera perl interpretará un arreglo (array) en lugar de una cadena (string)

Con Perl no tengo problemas para hacer lo mismo pero en lugar de usar AUTH LOGIN usar **AUTH PLAIN** usando la única cadena codifica del par usuario y contraseña

```
$ perl -MMIME::Base64 -e 'print encode_base64("\000usuario\@servidor.remoto.tld\000PaSsW0rD")'
AHphYmJpeEBjZW50cmFsLkTgqW3RyaXMuY29tADRNbXI1VG9wM0ZzbVF2S3RiOEVp
$ h=servidor.smtp.com
$ p=455
$ openssl s_client -connect $h:$p -starttls smtp
...
---
read R BLOCK
AUTH PLAIN AHphYmJpeEBjZW50cmFsLkTgqW3RyaXMuY29tADRNbXI1VG9wM0ZzbVF2S3RiOEVp
235 2.7.0 Authentication successful
```

Enlaces relacionados

- Verificar versiones y ciphers soportados de TLS, SSL
- Test SMTP with telnet or openssl

Aviso

CSF Firewall: añadir IPs al deny de forma definitiva.

Firewall y bloqueo definitivo para indeseables

A veces, trabajando ves que hay algunas IP que pertenecen a alguien, o incluso a un grupo de IP del mismo proveedor, que no para de aparecer en tus logs.

Además, si manda un correo al `abuse@` de l proveedor de servicios, se hace el loco y no te contesta.

Pues que mejor que banear la IP de forma definitiva.

CSF Firewall bloqueo permanente

Esta ahí, en los comentarios del `/etc/csf/csf.deny`, pero es una de los mas desconocidos del CSF.

```
# Note: If you add the text "do not delete" to the comments of an entry then
# DENY_IP_LIMIT will ignore those entries and not remove them
```

Maravilloso verdad?

Ya sólo queda banearle desde el shell.

```
❯ csf -d 179.43.128.0/18 "do not delete - Panama datacenter sin respuesta"
Adding 179.43.128.0/18 to csf.deny and iptables DROP...
csf: IPSET adding [179.43.128.0/18] to set [chain_DENY]
❯ cat /etc/csf/csf.deny | grep 179.43.128
179.43.128.0/18 # do not delete - Panama datacenter sin respuesta - Sat Aug 17 17:47:12 2024
```

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.

Certificados Letsencrypt sin servidor web o sin resolver en el servidor web

Como obtener el certificado Letsencrypt para un dominio que no resuelve aun en una migración o nuevo servidor

Algunas veces es necesario hacer pruebas antes de levantar una migración en un nuevo servidor en producción.

Para ello, deberemos de forma manual, obtener un nuevo certificado basándonos en el desafio llamado **challenge** en el que solicitaremos el certificado porque tenemos capacidad de administración de la zona DNS. También se conoce como **acme-dns-certbot**

Requisitos

En este doc presuponemos que:

- Tienes cierto nivel de usuario Linux.
- Que tienes instalado **certbot** con *snap*. Si no, acude a las Instrucciones de Cerbot
- Que tienes instalado **Python 3**, lo cual es ya lo mas común en una instalación de **Linux.**

Instalar acme-cerbot

Una vez instalado necesitamos descargar el script de phyton que nos permitirá trabajar con este tipo de desafío, o validación mediante DNS.

> Antes descargar nada, es buena práctica revisar el repositorio desde el que vamos a descargar el script. Antiguamente, no había forma salvo que conocieras un poco el programa y los sistemas implicados. Hoy día puedes usar si no alcanzas a esto, una chat de IA para que te verifique el scripts y te lo explique, como si fueras un novato en sistemas Linux y Python.

```
wget https://github.com/joohoi/acme-dns-certbot-joohoi/raw/master/acme-dns-auth.py
```

Lo hacemos ejecutable

```
chmod +x acme-dns-auth.py
```

Lo editamos para decirle que use Python 3

```
nano acme-dns-auth.py


#!/usr/bin/env python3
...
```

Una vez que hallamos realizado el cambio, movemos el fichero

```
sudo mv acme-dns-auth.py /etc/letsencrypt/
```

Configurar y usar acem-dns-cerbot

La cuestión es hora simple

```
sudo certbot certonly --manual --manual-auth-hook /etc/letsencrypt/acme-dns-auth.py --preferred-challenges dns --debug-challenges -d \*.tu-dominio -d tu-dominio
```

> Eso es una idea, basándonos en que tienes un * en tu zona dns. Pero puedes dejarlo en la simpleza del dominio normal, el que contiene *www* y otros como *mail*, etc. Pero el consejo es que **todos resuelvan a la ip**

Ejemplo ficticio que surge de tener todo lo que te pida respecto de la zona DNS del dominio solicitado.

Antes de darle a continuar **Press Enter to Continue** es evidente que ya has creado el registro en la zona del dominio, tal y como te solicitan, o de lo contrario fallará la generación del certificado Let's Encrypt.

```
certbot certonly --manual --manual-auth-hook /etc/letsencrypt/acme-dns-auth.py --preferred-challenges dns --
debug-challenges -d nodo1.midominio.tld
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for nodo1.midominio.tld
Hook '--manual-auth-hook' for nodo1.midominio.tld ran with output:
 Please add the following CNAME record to your main DNS zone:
 _acme-challenge.nodo1.midominio.tld CNAME cc94069f-6419-4c31-b079-d4408ec2bac6.auth.acme-dns.io.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Challenges loaded. Press continue to submit to CA.
Pass "-v" for more info about challenges.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Press Enter to Continue

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/nodo1.midominio.tld/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/nodo1.midominio.tld/privkey.pem
This certificate expires on 2025-08-11.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Con esto ya puedes configurar el sitio de manera temporal, hasta que resuelva en la maquina. Después lo suyo seria revocarlo y comenzar un proceso normal, basado en web.

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.