Lexislación e Seguridade Informática:

Práctica III. Protocolos Seguros y Auditorías de Seguridad

3.2.7-49.47

ssh lsi@10.11.49.47

1. ***Tomando como base de trabajo el SSH pruebe sus diversas utilidades:***

   a) ***Abra un shell remoto sobre SSH y analice el proceso que se realiza. Configure su fichero ssh_known_hosts para dar soporte a la clave pública del servidor.***

ssh -v lsi@10.11.49.58: modo verbose, pone lo que hace el ssh.

Creamos la clave pública: **ssh-keygen -l**

```
lsi@debian:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lsi/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/lsi/.ssh/id_rsa
Your public key has been saved in /home/lsi/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:VQvCHtmhRBC8uQ1TaEfFGixvDuYKkaqFfRWjqB2xBQA lsi@debian
The key's randomart image is:
+---[RSA 3072]----+
|E...  .oO=+..    |
|   . . B.Oooo .  |
|    * o @.+. .   |
|   * . O =.      |
| * o + OS        |
|+ = . o o        |
|.. o .           |
|.   .            |
|                 |
+----[SHA256]-----+
lsi@debian:~$ ssh-keygen -l
Enter file in which the key is (/home/lsi/.ssh/id_rsa):
3072 SHA256:VQvCHtmhRBC8uQ1TaEfFGixvDuYKkaqFfRWjqB2xBQA lsi@debian (RSA)
```

Añadimos las claves públicas de nuestros compañeros de prácticas:

ssh-keyscan 10.11.49.55 >> /etc/ssh/ssh_known_hosts

ssh-keyscan 10.11.49.58 >> /etc/ssh/ssh_known_hosts

Para ver que funciona en el $HOME/.ssh/known_hosts borramos todo. Si después sale el mensaje de fingerprinting está mal.

Primero se mira si la pública está en /etc/ssh/ssh_known_hosts, luego en $HOME/.ssh/known_hosts y si no está en ninguno, la inserta.

b) Haga una copia remota de un fichero utilizando un algoritmo de cifrado determinado. Analice el proceso que se realiza.

Ver los diferentes algoritmos de cifrado: **ssh -Q cipher**

Local a Remoto:

scp -c aes128-ctr enviarAdrian.txt lsi@10.11.49.55:/home/lsi

Remoto a Local:

scp -c aes128-ctr lsi@10.11.49.55:/home/lsi/enviarDavid.txt /home/lsi

c) Configure su cliente y servidor para permitir conexiones basadas en un esquema de autenticación de usuario de clave pública.

ssh-keygen -t rsa (hacer todo como usuario lsi en este apartado)

(Va a pedir frase de paso dos veces, las dejamos en blanco)

Comprobamos en ~/.ssh/ que existan los ficheros id_rsa y id_rsa.pub

Después hacemos lo mismo con **ecdsa** y **ed25519**

Le enviamos los 3 .pub a nuestro compañero (el comando lo mete en el $HOME/.ssh/authorized_keys):

ssh-copy-id -i $HOME/.ssh/id_rsa.pub lsi@10.11.49.55
ssh-copy-id -i $HOME/.ssh/id_ecdsa.pub lsi@10.11.49.55
ssh-copy-id -i $HOME/.ssh/id_ed25519.pub lsi@10.11.49.55

ssh-copy-id -i $HOME/.ssh/id_rsa.pub lsi@10.11.49.58
ssh-copy-id -i $HOME/.ssh/id_ecdsa.pub lsi@10.11.49.58
ssh-copy-id -i $HOME/.ssh/id_ed25519.pub lsi@10.11.49.58

Si no nos pide contraseña al hacer ssh funciona

d) Mediante túneles SSH securice algún servicio no seguro.

Securizaremos el puerto 80:

Antes de nada: **a2dismod evasive** y **a2dismod security2** para desactivar el modsecurity (después systemctl restart apache2)

ssh -P -L 10080:10.11.49.55:80 lsi@10.11.49.55

Mientras la conexión ssh funcione, seremos capaces de hacer peticiones al servidor apache instalado en la máquina de nuestro compañero, y redireccionar todo ese tráfico del puerto 80 por nuestra conexión ssh.

En otra terminal: **lynx http://localhost:10080** y se nos abre una página de apache

e) ***Exporte" un directorio y "móntelo" de forma remota sobre un túnel SSH.***

NO ENTRA

f) ***PARA PLANTEAR DE FORMA TEÓRICA.: Securice su sevidor considerando que únicamente dará servicio ssh para sesiones de usuario desde determinadas IPs.***

NO ENTRA

2. ***Tomando como base de trabajo el servidor Apache2***

a) ***Configure una Autoridad Certificadora en su equipo.***

Instalamos openssl: **apt install openssl**

Nos movemos de directorio: **cd /usr/lib/ssl/misc**

Creamos la entidad certificadora: **./CA.pl -newca**

Frase de paso: frasedepaso

Country name: ES

State: A-Coruña

Locality: A-Coruña

Organization name: lsi

Organizational unit name: lsi

Common Name: ramallal

Email: david.ramallal@udc.es

Challenge password: contraña de la maquina

Optional Company name: []

Con esto, ya somos una Autoridad Certificadora, y se habrán creado dos ficheros:

/usr/lib/ssl/misc/demoCA/cacert.pem // Certificado con clave pública de CA

/usr/lib/ssl/misc/demoCA/private/cakey.pem //Clave privada cifrada con frase de paso

```
Signature ok
Certificate Details:
        Serial Number:
            77:63:89:47:2a:d5:63:10:ee:5e:b7:ca:46:1b:7e:98:0b:45:79:cd
        Validity
            Not Before: Nov 28 17:20:42 2022 GMT
            Not After : Nov 27 17:20:42 2025 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = A-Corula\08\08\C3\B1a
            organizationName          = lsi
            organizationalUnitName    = lsi
            commonName                = ramallal
            emailAddress              = david.ramallal@udc.es
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                4C:08:2C:B7:CF:DE:10:91:BB:C8:C3:38:D4:CB:8E:9C:31:70:98:4C
            X509v3 Authority Key Identifier:
                keyid:4C:08:2C:B7:CF:DE:10:91:BB:C8:C3:38:D4:CB:8E:9C:31:70:98:4C

            X509v3 Basic Constraints: critical
                CA:TRUE
Certificate is to be certified until Nov 27 17:20:42 2025 GMT (1095 days)

Write out database with 1 new entries
Data Base Updated
⟹ 0
====
CA certificate is in ./demoCA/cacert.pem
```

b) Cree su propio certificado para ser firmado por la Autoridad Certificadora. Bueno, y fírmelo.

Creamos el certificado (en el mismo directorio):

./CA.pl -newreq-nodes

Pide los mismos campos que antes (esta vez la frase de paso en blanco, para evitar problemas)

Se generan dos ficheros (newkey.pem, newreq.pem)

Firmamos el certificado (la firma digital es con la clave privada de la CA):

./CA.pl -sign

La frase de paso es la de la CA (frasedepaso)

Se crea el newcert.pem

/usr/lib/ssl/misc/newkey.pem // Clave privada

/usr/lib/ssl/misc/newcert.pem // Certificado con la pública firmado

Al firmar se creará un nuevo archivo **newcert.pem**, que junto a la clave privada newkey.pem, es lo que le tendremos que dar a nuestro compañero (por medio de scp, por ejemplo).

c) Configure su Apache para que únicamente proporcione acceso a un determinado directorio del árbol web bajo la condición del uso de SSL. Considere que si su la clave privada está cifrada en el proceso de arranque su máquina le solicitará la correspondiente frase de paso, pudiendo dejarla inalcanzable para su sesión ssh de trabajo.

Habilitamos ssl:

a2enmod ssl

a2ensite default-ssl

systemctl restart apache2

En **/etc/apache2/ports.conf** metemos (si no está ya)

Listen 80

Listen 443

Somos 3: uno es servidor web y dos clientes (uno de los clientes hace de CA)

El servidor: **chmod 777 /usr/lib/ssl/misc**

La CA le pasa al servidor el newcert.pem y le newkey.pem y lo mete en /usr/lib/ssl/misc:

scp -c aes128-ctr /usr/lib/ssl/misc/newcert.pem lsi@10.11.49.47:/usr/lib/ssl/misc/newcert_.pem

scp -c aes128-ctr /usr/lib/ssl/misc/newkey.pem lsi@10.11.49.47:/usr/lib/ssl/misc/newkey_.pem

El servidor: **chmod 755 /usr/lib/ssl/misc**

En el fichero **/etc/apache2/sites-available/default-ssl.conf** (fichero de conf de https) tenemos por defecto:

SSLEngine ON

SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem

SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

El **servidor**:

SSLCertificateFile /usr/lib/ssl/misc/newcert_.pem

SSLCertificateKeyFile /usr/lib/ssl/misc/newkey_.pem

Reiniciamos: **systemctl restart apache2**

Los clientes meten en el /etc/hosts la ip con el common name

A continuación la CA pasa su cacert a los otros 2 y los 3 la meten en ca-certificates:

cp /usr/lib/ssl/misc/demoCA/cacert.pem /usr/local/share/ca-certificates/cacert.crt

update-ca-certificates

Desde los clientes nos conectamos al servidor: lynx https://common-name

3. ***Tomando como base de trabajo el openVPN deberá configurar una VPN entre dos equipos virtuales del laboratorio que garanticen la confidencialidad entre sus comunicaciones.***

Instalamos openssl (ya estaba de otro apartado anterior) y openvpn

apt install openvpn

Hacemos **lsmod | grep tun**, si no nos aparece hacemos lo siguiente:

modprobe tun

echo tun >> /etc/modules

Yo soy cliente

Servidor:

Nos movemos a la carpeta (**cd /etc/openvpn**)

openvpn –genkey –secret clave.key

nano tunel.conf

local 10.11.49.55

remote 10.11.49.47

dev tun1

port 5555

comp-lzo

user nobody

ping 15

ifconfig 172.160.0.1 172.160.0.2

secret /etc/openvpn/clave.key

systemctl enable openvpn

systemctl start openvpn

Cliente:

Nos movemos a la carpeta (**cd /etc/openvpn**)

nano tunel.conf

local 10.11.49.47

remote 10.11.49.55

dev tun1

port 5555

comp-lzo

user nobody

ping 15

ifconfig 172.160.0.2 172.160.0.1

secret /etc/openvpn/clave.key

systemctl enable openvpn

systemctl start openvpn

Le hacemos ping a 172.160.0.2 y a 172.160.0.1 para comprobar que funciona

4. ***EN LA PRÁCTICA 1 se configuró una infraestructura con servidores y clientes NTP. Modifique la configuración para autenticar los equipos involucrados.***

NO ENTRA

5. ***EN LA PRÁCTICA 1 se instalaron servidores y clientes de log. Configure un esquema que permita cifrar las comunicaciones.***

NO ENTRA

6. ***En este punto, cada máquina virtual será servidor y cliente de diversos servicios (NTP, syslog, ssh, web, etc.). Configure un "firewall stateful" de máquina adecuado a la situación actual de su máquina.***

```
#!/bin/bash

#BORRAR IPTABLES ANTERIORES
/sbin/iptables -F
/sbin/iptables -X
/sbin/iptables -t nat -F

#DROPEAMOS TODO
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP

#BORRAR IP6TABLES ANTERIORES
/sbin/ip6tables -F
/sbin/ip6tables -X
/sbin/ip6tables -t nat -F

#DROPEAMOS TODO
/sbin/ip6tables -P INPUT DROP
/sbin/ip6tables -P OUTPUT DROP
/sbin/ip6tables -P FORWARD DROP

#Gardamos os logs
/sbin/iptables -A INPUT -j LOG

YO=10.11.49.47
COMPI1=10.11.49.55
COMPI2=10.11.49.58
```

YO51=10.11.51.47

COMPI1_51=10.11.51.55

COMPI2_51=10.11.51.58

YOIPV6=2002:a0b:312f::1

COMPI1_IPV6=2002:a0b:3137::1

COMPI2_IPV6=2002:a0b:313a::1

YOVPN=172.160.0.2

COMPI1VPN=172.160.0.1

#---------- PERMITIMOS ----------

#Aceptamos todo el tráfico de Localhost

/sbin/iptables -A INPUT -i lo -j ACCEPT

/sbin/iptables -A OUTPUT -o lo -j ACCEPT

/sbin/iptables -A INPUT -i ens33 -p ipv6 -j ACCEPT

/sbin/iptables -A OUTPUT -o ens33 -p ipv6 -j ACCEPT

#Aceptamos cualquier paquete que provenga de conexiones ya establecidas

/sbin/iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

/sbin/iptables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

/sbin/ip6tables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

/sbin/ip6tables -A OUTPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

#SSH (ens33)

/sbin/iptables -A INPUT -s $COMPI1 -d $YO -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI2 -d $YO -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI1 -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI2 -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

#SSH (ens34)

/sbin/iptables -A INPUT -s $COMPI1_51 -d $YO51 -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI2_51 -d $YO51 -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO51 -d $COMPI1_51 -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO51 -d $COMPI2_51 -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

#HTTP

/sbin/iptables -A INPUT -s $COMPI1 -d $YO -p tcp --dport http -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI2 -d $YO -p tcp --dport http -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI1 -p tcp --dport http -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI2 -p tcp --dport http -m conntrack --ctstate NEW -j ACCEPT

#HTTPS

/sbin/iptables -A INPUT -s $COMPI1 -d $YO -p tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI2 -d $YO -p tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI1 -p tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI2 -p tcp --dport 433 -m conntrack --ctstate NEW -j ACCEPT

#Eduroam

/sbin/iptables -A INPUT -s 10.20.32.0/21 -d $YO -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s 10.30.8.0/21 -d $YO -p tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT

#IPV6

/sbin/ip6tables -A INPUT -s $COMPI1_IPV6 -m conntrack --ctstate NEW -j ACCEPT

/sbin/ip6tables -A OUTPUT -d $COMPI1_IPV6 -m conntrack --ctstate NEW -j ACCEPT

/sbin/ip6tables -A INPUT -s $COMPI2_IPV6 -m conntrack --ctstate NEW -j ACCEPT

/sbin/ip6tables -A OUTPUT -d $COMPI2_IPV6 -m conntrack --ctstate NEW -j ACCEPT

#ICMP

/sbin/iptables -A INPUT -s $COMPI1 -d $YO -p icmp -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI2 -d $YO -p icmp -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI1VPN -p icmp -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI1 -p icmp -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI2 -p icmp -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -d $COMPI1VPN -p icmp -m conntrack --ctstate NEW -j ACCEPT

#TunelVPN

/sbin/iptables -A INPUT -s $COMPI1 -d $YO -p udp --dport 5555 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d $COMPI1 -p udp --sport 5555 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI1VPN -d $YOVPN -p tcp --dport 5555 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YOVPN -d $COMPI1VPN -p tcp --sport 5555 -m conntrack --ctstate NEW -j ACCEPT

#NTP (servidor)

/sbin/iptables -A INPUT -s $COMPI1 -d $YO -p udp --dport 123 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A INPUT -s $COMPI2 -d $YO -p udp --dport 123 -m conntrack --ctstate NEW -j ACCEPT

#Syslog (cliente)

/sbin/iptables -A OUTPUT -s $YO -d $COMPI1 -p tcp --dport 514 -m conntrack --ctstate NEW -j ACCEPT

#Nessus

/sbin/iptables -A INPUT -s 10.30.8.0/21 -d $YO -p tcp --dport 8834 -m conntrack --ctstate NEW -j ACCEPT

#Splunk

/sbin/iptables -A INPUT -s 10.30.8.0/21 -d $YO -p tcp --dport 8000 -m conntrack --ctstate NEW -j ACCEPT

#DNS

/sbin/iptables -A OUTPUT -s $YO -d 10.8.12.49 -p tcp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d 10.8.12.49 -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d 10.8.12.47 -p tcp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d 10.8.12.47 -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d 10.8.12.50 -p tcp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d 10.8.12.50 -p udp --dport 53 -m conntrack --ctstate NEW -j ACCEPT

#APT

/sbin/iptables -A OUTPUT -s $YO -d deb.debian.org -m conntrack --ctstate NEW -j ACCEPT

/sbin/iptables -A OUTPUT -s $YO -d security.debian.org -m conntrack --ctstate NEW -j ACCEPT

#----------POR SEGURIDAD----------

sleep 30

Limpiar reglas existentes

/sbin/iptables -F

/sbin/iptables -X

/sbin/ip6tables -F

/sbin/ip6tables -X

Restaurar políticas por defecto

/sbin/iptables -P INPUT ACCEPT

/sbin/iptables -P FORWARD ACCEPT

/sbin/iptables -P OUTPUT ACCEPT

/sbin/ip6tables -P INPUT ACCEPT

/sbin/ip6tables -P FORWARD ACCEPT

/sbin/ip6tables -P OUTPUT ACCEPT

7. ***Instale el SIEM splunk en su máquina. Sobre dicha plataforma haga los siguientes puntos:***

Descargamos de Teams la versión 8.2.1 de Splunk y, desde la máquina, le cambiamos los permisos y la instalamos.

Hacemos: **/opt/splunk/bin/splunk enable boot-start**

Ponemos el usuario y password que queramos (lsi y nuestra contraseña por ejemplo)

Activamos splunk: **systemctl start splunk**

Para entrar, desde nuestro portátil ponemos en el navegador **http://10.11.49.47:8000**

En **/opt/splunk/etc/system/default/server.conf** cambiamos la variable minFreeSpace a 500 (si no lo hacemos podemos matar la máquina por ocupación de disco)

a) Genere una query que visualice los logs internos del splunk

En el menú de Splunk elegimos la opción “Search & Reporting”.

En el buscador escribimos la query:

index = _internal

Esta query nos permite ver los logs internos de splunk.

b) Cargué el fichero /var/log/apache2/access.log y el journald del sistema y visualícelos

En el menú de Splunk elegimos las opciones Add Data > Monitor

Cargamos los ficheros **/var/log/syslog, /var/log/apache2/access.log** desde la opción “Files & Directories” y el **journald** desde la opción “Systemd Journald Input for Splunk”.

Ahora podemos realizar consultas en el buscador:

host = “debian” -> Saca los logs de la máquina llamada “debian”

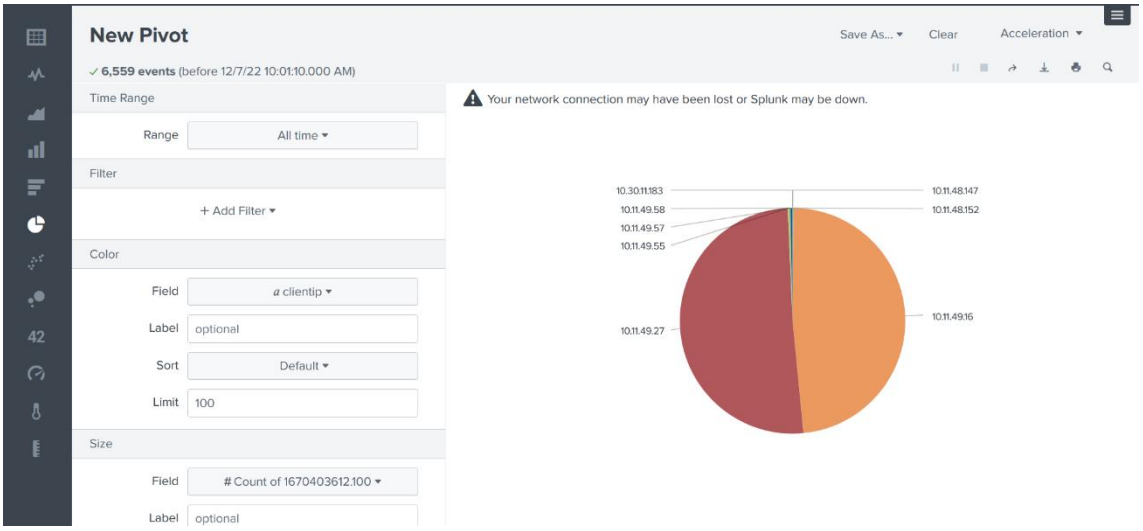
host = “debian” source = "/var/log/apache2/access.log" -> saca los logs de ese fichero del host “debian”

host = "debian" source="journald://journald splunk" -> saca logs journald

c) ***Obtenga las IPs de los equipos que se han conectado a su servidor web (pruebe a generar algún tipo de gráfico de visualización), así como las IPs que se han conectado un determinado día de un determinado mes***

host = “debian” source = "/var/log/apache2/access.log" date_month = "december" date_mday = “5” -> saca los logs de las conexiones a mi servidor el 5 de diciembre.

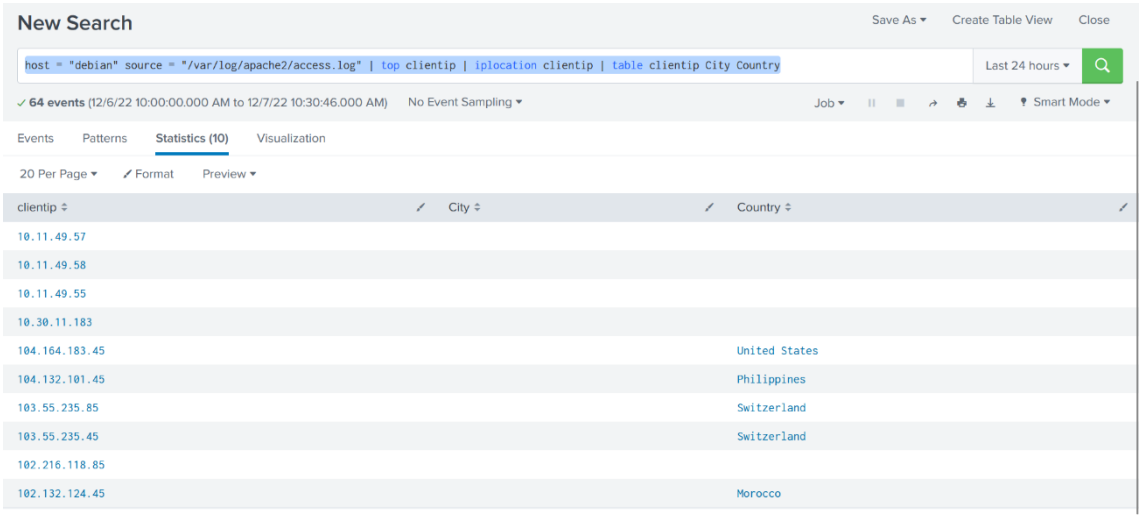
host = “debian” source = "/var/log/apache2/access.log" date_month = "december" date_mday = “5” “10.11.49.58” -> saca los logs de las conexiones a mi servidor desde la máquina de mi compañero el 5 de diciembre.



d) ***Trate de obtener el país y región origen de las IPs que se han conectado a su servidor web y si posible sus coordenadas geográficas.***

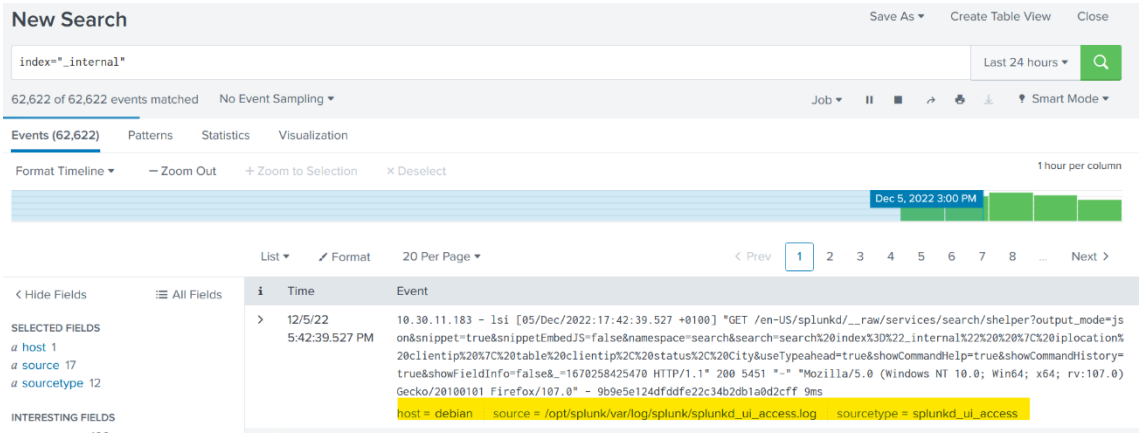
host = "debian" source = "/var/log/apache2/access.log" | top clientip | iplocation clientip | table clientip City Country (hay que crear el campo clientip si no se detecta automáticamente)

Las máquinas de nuestros compañeros no registran la localización así que meteremos manualmente en el fichero /var/log/apache2/access.log ips de otros países que encontremos por internet.



e) Obtenga los hosts origen, sources y sourcestypes.

Al hacer una consulta podemos ver los tres campos:



f) ¿cómo podría hacer que splunk haga de servidor de log de su cliente?

(SOLO TEÓRICO, NO HACER)

En el menú: Add Data > Monitor elegimos la opción “TCP / UDP”

Activo TCP poniendo el puerto 514 y la ip de las máquinas que quiero que les de permiso a meterme log.

En la máquina del compañero quitas el Rsyslog del puerto 514 (para que no haya conflicto).

Ahora los logs ya no los recoge Rsyslog, los pilla directamente Splunk.

8. ***Ejecute la utilidad de auditoría de seguridad lynis en su sistema y trate de identificar las acciones de securización detectadas así como los consejos sobre las que se deberían contemplar.***

NO ENTRA

9. ***EN LA PRÁCTICA 2 se obtuvo un perfil de los principales sistemas que conviven en su red, puertos accesibles, fingerprinting, paquetería de red, etc. Seleccione un subconjunto de máquinas del laboratorio de prácticas y la propia red. Elabore el correspondiente informe de análisis de vulnerabilidades. Puede utilizar como apoyo al análisis la herramienta Nessus Essentials (disponible para educación en https://www.tenable.com/tenable-for-education/nessus-essentials bajo registro para obtener un código de activación) para su instalación en la máquina debian de prácticas. Como opción alternativa, también podría instalar Greenbone Vulnerability Management (GVM). Como referencia-plantilla puede utilizar:***
    - ***Writing a Penetration Testing Report del SANS (SysAdmmin Audit, Networking and Security) Institute. Muestra las etapas o fases del desarrollo de un "report", describe el formato del "report" y finaliza con un ejemplo. http://www.sans.org/reading-room/whitepapers/bestprac/writing-penetration-testing-report-33343?show=writing-penetration-testing-report-33343&cat=bestprac***
    - ***Plantilla de vulnerabilityassessment.co.uk. http://www.vulnerabilityassessment.co.uk/report%20template.html***

    En la web https://www.tenable.com/tenable-for-education/nessus-essentials nos registramos con la cuenta de la UDC y nos dan una licencia.

    Descargamos para debian 9.10 y lo mandamos a nuestra máquina.

    Instalamos: **apt install ./Nessus-10.4.1-debian9_amd64.deb**

    systemctl start nessusd.service

    Desde fuera de la máquina abrimos un navegador y ponemos: **https://10.11.49.47:8834**

    Elegimos Nessus Essentials y creamos usuario y password.

    Esperamos a que se descarguen todos los plugins (tarda alrededor de una hora)

Name	Schedule	Last Scanned ▾
Router	On Demand	December 4 at 12:08 PM
Mi máquina después de actualizar	On Demand	December 3 at 7:08 PM
Escaneo de prueba	On Demand	December 3 at 12:18 PM