Antes que nada debemos instalar el Forwarder de Splunk en el servidor Debian donde está instalado Apache. Lo hacemos ejecutando como root:

```
curl -sL https://raw.githubusercontent.com/nipegun/d-scripts/refs/heads/master/SoftInst/ParaCLI/Splunk-UniversalForwarder-Instalar.sh | bash
```

Al concluir la instalación debemos indicar al forwarder la IP y el puerto de escucha de servidor SIEM Splunk al que vamos a reenviar los logs. Lo hacemos ejecutando como root:

```
/opt/splunkforwarder/bin/splunk add forward-server 192.168.1.10:9997
```

Nos pedirá un usuario y contraseña. El usuario es **admin** y la contraseña por defecto al instalar Splunk Forwarder desde mi script de Github es **UsuarioX**.

A continuación, y para aplicar los cambios, ejecutamos como root:

```
/opt/splunkforwarder/bin/splunk restart
```

Ahora deberemos indicar que logs queremos enviar al SIEM. Si hemos instalado apache y no hemos tocado la ubicación por defecto de los logs, estos estarán en **/var/log/apache2/access/log** y **/var/log/apache2/error.log**, con lo que monitorizaremos ambos logs ejecutando como root:

```
/opt/splunkforwarder/bin/splunk add monitor /var/log/apache2/access.log
```

Ingresamos el usuario y la contraseña: **admin** y **UsuarioX**.

```
/opt/splunkforwarder/bin/splunk add monitor /var/log/apache2/error.log
```

Aquí, al haberlos puesto unos segundos antes, no nos pedirá ni usuario ni contraseña. Entonces, reiniciamos el forwarder ejecutando como root:

```
/opt/splunkforwarder/bin/splunk restart
```

A esta altura ya estaríamos enviando logs de la Web al SIEM. El problema es que, si no hemos configurado que los logs se envíen en formato combinado (u otro formato más detallado) los logs web que lleguen al SIEM lo estarán haciendo con el sourcetype **access-too_small**, y ese formato hace llegar muy pocos campos de datos a los logs del SIEM. Para cambiar esto deberemos indicar al forwarder que envíe los logs al SIEM en formato combinado. Esto lo hacemos ejecutando como root:

```
mkdir -p /opt/splunkforwarder/etc/apps/apache/local/
echo '[monitor:///var/log/apache2/access.log]'  > /opt/splunkforwarder/etc/apps/apache/local/inputs.conf
echo 'sourcetype = access_combined'            >> /opt/splunkforwarder/etc/apps/apache/local/inputs.conf
```

Si no tenemos tráfico suficiente porque acabamos de instalar el servidor Web, podemos utilizar este script de python para simularlo.

Ahora nos queda encontrar algunas consultas interesantes que podemos meterle a Splunk para obtener buena info de lo que está ocurriendo en el servidor:

CONSULTAS PARA WEB

Contar y agrupar las visitas por el tipo de navegador con el que se intenta visitar la web:

```
host="debianserver" source="/var/log/apache2/access.log" | stats count by useragent
```

Mostrar la cantidad de status 200 y status 400 que hay registrados:

```
host="debianserver" source="/var/log/apache2/access.log" | search (status=200 OR status=404) | stats count by status
```

URLs que se solicitan:

```
host="debianserver" source="/var/log/apache2/access.log" | stats count by file | sort count desc
```

Horas a las que más se accede:

```
host="debianserver" source="/var/log/apache2/access.log" | eval hour=strftime(_time, "%H") | stats count by hour | sort hour
```

Acceso por países:

```
host="debianserver" source="/var/log/apache2/access.log" | stats count by clientip | iplocation clientip | sort by Country | stats count by Country
```

IPs que más acceden:

```
host="debianserver" source="/var/log/apache2/access.log" | stats count by clientip | sort desc
```

Consulta para ubicar en el mapa las IPs que acceden a la web:

```
host="debianserver" source="/var/log/apache2/access.log" | iplocation clientip | stats count by City Country lat lon | geostats latfield=lat longfield=lon count
```