Logs

Los archivos de registro (*logs*) son un conjunto de registros que permiten a los administradores realizar un seguimiento de los eventos importantes. Contienen mensajes sobre equipos, servicios, aplicaciones, ... Es fundamental para las organizaciones monitorizar el funcionamiento de sus sistemas y disponer de los logs para poder hacer un seguimiento de las incidencias.

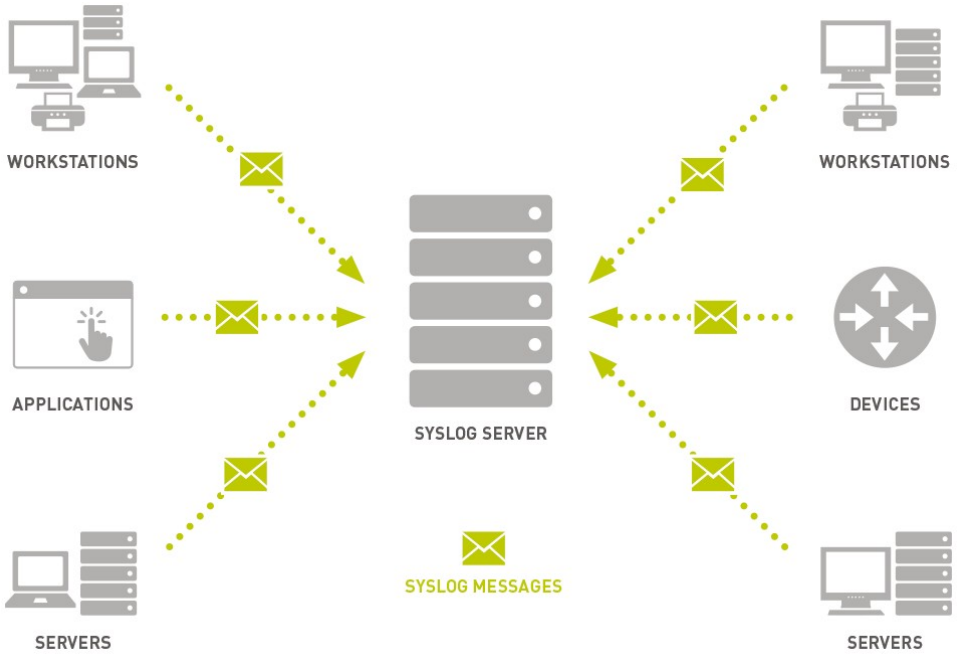
En el caso de análisis forense, los logs se pueden agrupar en tres grupos:

- Registros de dispositivos de red (enrutadores, conmutadores, ...)
- Registros de dispositivos de seguridad (IDS, firewalls, proxies, NGFW, WAF, ...)
- Registros desde el lado del punto final (servidor, escritorio, ...)

En sistemas Windows está el visor de eventos donde se puede acceder a los logs del sistemas, seguridad y aplicaciones. En sistemas Linux los archivos de log están normalmente localizado en la carpeta /var/log:

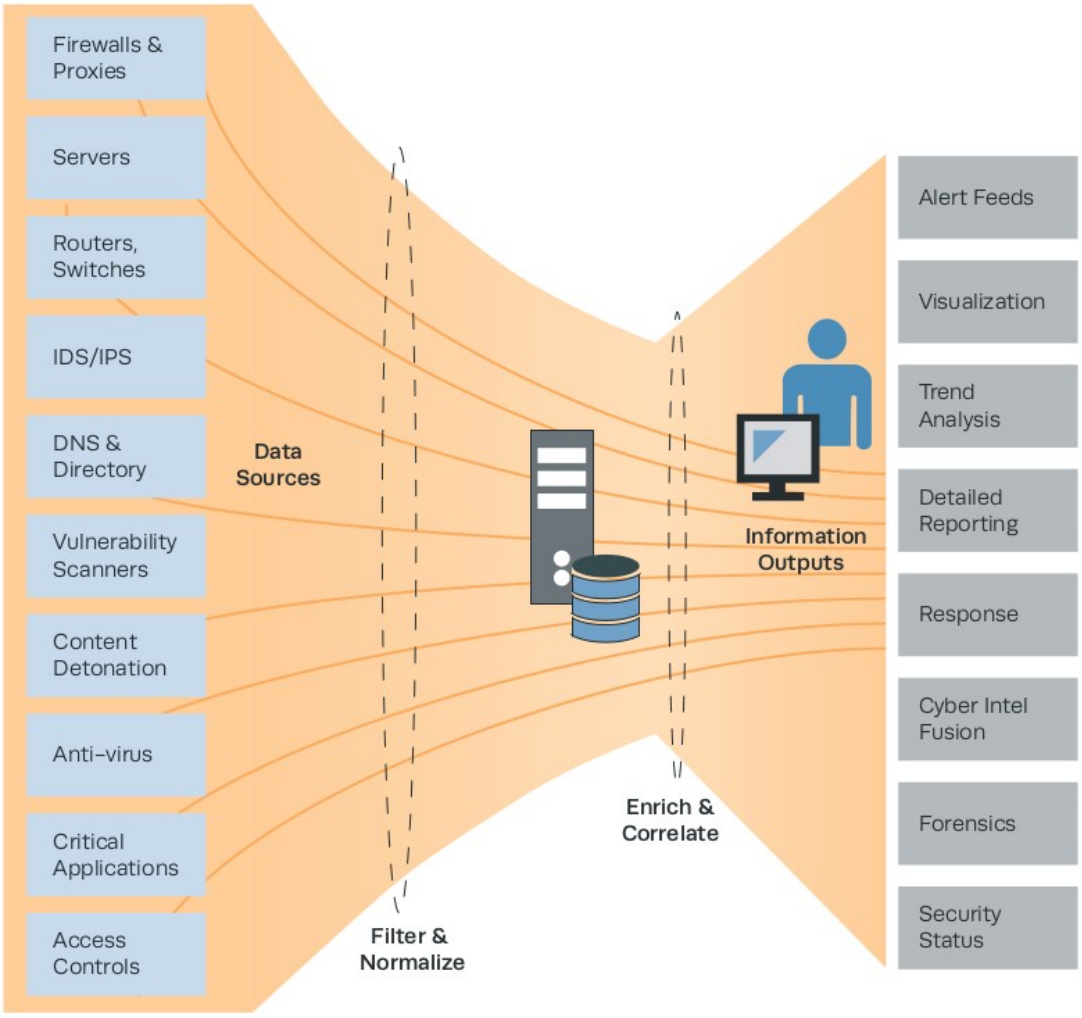
- /var/log/message o /var/log/syslog: almacena mensajes informativos y no críticos del sistema.
- /var/log/auth.log: eventos relativos a la autenticación y autorización de usuarios.
- /var/log/dpkg.log: registra que paquetes se instalaron y/o desinstalaron en el sistema.
- /var/log/kern.log: mensajes de errores y advertencias relacionados con el kernel.
- /var/log/boot.log: información sobre el arranque.
- /var/log/dmesg: información relacionada con los dispositivos de hardware y sus controladores. A medida que el kernel detecta dispositivos de hardware físicos asociados con el servidor durante el proceso de arranque, captura el estado del dispositivo, los errores de hardware y otros mensajes genéricos.
- /var log/utmp or /var/log/wtmp: registro de todas las conexiones y desconexiones del sistema. Cada vez que un usuario se conecta, se inicia o se apaga el sistema.

Los logs se pueden guardar en los equipos de forma local pero es recomendable centralizarlos. La siguiente imagen ejemplifica el concepto de centralización, ya que todos los dispositivos de interés están configurados para enviar mediante syslos sus registros a un servidor central.



De vital importancia es que los equipos de la organización estén sincronizados para que a la hora de analizarlos, sus marcas de tiempo (*timestamps*) nos sean de ayuda para correlacionar eventos y no un estorbo. La sincronización de los equipos puede hacer usando el servicio NTP (Network Time Protocol) y en caso de analizar registros de equipos situados en zonas horarias diferentes (p.e. equipos situados en la peninsula ibérica, en Canarias y en New York), lo recomendado es trabajar con fechas en formato UTC.

Si el grado de madurez en seguridad de la organización es suficientemente alto, es habitual disponer de agentes en los dispositivos que envían registros al SIEM para que los analistas puedan analizarlos en un punto central:



La centralización de logs es también un mecanismo de seguridad frente a la eliminación de registros por parte de los atacantes, en su afán por borrar pistas de sus actividades.

Procedimiento general

1. Identificar qué fuentes de registro y herramientas automatizadas se pueden usar durante el analisis.
2. Copiar los registros en una sola ubicación para revisarlos.
3. Minimizar el "ruido" eliminando las entradas de registro repetitivas y rutinarias, para después confirmar que son benignas.
4. Determinar si puede confiar en las marcas de tiempo de los registros; considerar los husos horarios.
5. Centrarse en cambios recientes, fallas, errores, cambios de estado, acceso y eventos de administración, y otros eventos inusuales para su ambiente.

6. Ir hacia atrás en el tiempo para reconstruir las acciones después y antes del incidente.
7. Correlacionar actividades en diferentes registros para obtener una visión completa
8. Desarrollar teorías sobre lo ocurrido; explorar registros para confirmar o desaprobarlos.

Análisis manual de log

El uso de algunos comandos va a permitir mejorar sustancialmente el análisis de los logs cuando se hace de forma manual. Los siguientes comandos linux son fundamentales para el análisis de logs y archivos del sistema:

- wc
- cat, less, more
- tail, head
- grep
- cut
- sort
- uniq
- awk
- sed
- man --> ayuda del comando cone explicación de sus opciones

Hay que tener presente que cada fichero de log tiene un formato de línea determinado, por lo que es necesario revisar su estructura y conocer el significado y ubicación de los campos así como la información que proporcionan; para poder proceder a su procesamiento y análisis. Por ejemplo, usando cat, head o tail se puede ver la estructura de un fichero de log típico de un servidor web Apache:

```
manuel@x99:~$ tail -n 1 access_log.txt
70.194.129.34 - - [25/Apr/2013:15:55:42 -0700] "GET / HTTP/1.1" 200 4023 "-" "Mozilla/5.0 (Linux; U; Android 4.1.2; en-us; SCH-I535 Build/JZO54K) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30" "www.random-site.com"
```

En la documentación del fabricante se puede obtener una explicación de cada elemento de la línea:

%h	%l	%u	%t	"%r"	%>s	%b	"%{Referer}i"	"%{User-agent}i"
IP cliente	-	usuario	fecha petición	Request	Status	tamaño respuesta sin cabeceras	Referer	UserAgent

A partir de ahí, viendo los campos y los elementos delimitadores de los mismos se puede proceder a usar comandos para su análisis.

Nº de líneas de un fichero

Contar el nº de líneas del fichero nmap.txt:

```
manuel@x99:~$ wc -l nmap.txt
49 nmap.txt
```

Seleccionar líneas

Mostrar las líneas con la palabra *open* presentes en el fichero nmap.txt:

```
manuel@x99:~$ grep open nmap.txt
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Windows Server 2003 3790 microsoft-ds
```

```
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
8089/tcp open  ssl/http     Splunkd httpd
```

Guardar las líneas con la palabra *open* en un fichero llamado open.txt

Se redirigen las líneas seleccionadas a otro fichero usando una redirección >:

```
manuel@x99:~$ grep open nmap.txt > open.txt
manuel@x99:~$ cat open.txt
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Windows Server 2003 3790 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
8089/tcp open  ssl/http     Splunkd httpd
```

También se puede hacer con el comando tee y una tubería (*pipe*) |:

```
manuel@x99:~$ grep open nmap.txt | tee open.txt
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Windows Server 2003 3790 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
8089/tcp open  ssl/http     Splunkd httpd
manuel@x99:~$ cat open.txt
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Windows Server 2003 3790 microsoft-ds
1025/tcp open  msrpc        Microsoft Windows RPC
1026/tcp open  msrpc        Microsoft Windows RPC
8089/tcp open  ssl/http     Splunkd httpd
```

Contar el número de puertos abiertos:

Con la opción -c de grep o usando una pipe y el comando wc -l

```
manuel@x99:~$ grep open nmap.txt | wc -l
6
manuel@x99:~$ grep -c open nmap.txt
6
```

Mostrar los puertos abiertos (*open*) que no estén asociados a RPC:

La salida del primer grep *alimenta* la entrada del segundo grep gracias a la tubería |:

```
manuel@x99:~$ grep open nmap.txt | grep -v RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Windows Server 2003 3790 microsoft-ds
8089/tcp open  ssl/http     Splunkd httpd
```

Mostrar en Kali los usuarios del fichero *passwd* que tengan una shell que no sea ni del tipo *nologin* ni *false*

Se puede usar la opción -e o expresiones regulares:

```
manuel@x99:~$ grep -v -e nologin -e false passwd
root:x:0:0:root:/root:/usr/bin/zsh
sync:x:4:65534:sync:/bin:/bin/sync
postgres:x:119:124:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
kali:x:1000:1000:kali,,,:/home/kali:/usr/bin/zsh
manuel@x99:~$ grep -v -E "(nologin|false)" passwd
root:x:0:0:root:/root:/usr/bin/zsh
sync:x:4:65534:sync:/bin:/bin/sync
postgres:x:119:124:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
kali:x:1000:1000:kali,,,:/home/kali:/usr/bin/zsh
```

Extracción de información por campos

Mostrar los nombres de usuario del fichero *passwd*

Usando el comando cut con la opción -d para indicar el delimitador del campo y la opción -f para indicar el/los

campo/s a mostrar.

```
manuel@x99:~$ cut -d ":" -f 1 passwd
root
daemon
bin
sys
sync
...
```

Mostrar por pantalla los usuarios del fichero *passwd* junto con su directorio *home* y su *shell*

```
manuel@x99:~$ cut -d ":" -f 1,6,7 passwd
root:/root:/usr/bin/zsh
daemon:/usr/sbin:/usr/sbin/nologin
bin:/bin:/usr/sbin/nologin
sys:/dev:/usr/sbin/nologin
sync:/bin:/bin/sync
...
```

Mostrar por pantalla los usuarios del fichero *passwd* junto con su directorio *home* y su *shell* usando awk

```
manuel@x99:~$ awk -F ":" '{print $1,$6,$7}' passwd
root /root /usr/bin/zsh
daemon /usr/sbin /usr/sbin/nologin
bin /bin /usr/sbin/nologin
sys /dev /usr/sbin/nologin
sync /bin /bin/sync
...
```

Igual que el anterior pero separando la información con ":"

```
manuel@x99:~$ awk -F ":" '{print $1":"$6":"$7}' passwd
root:/root:/usr/bin/zsh
daemon:/usr/sbin:/usr/sbin/nologin
bin:/bin:/usr/sbin/nologin
sys:/dev:/usr/sbin/nologin
...
```

Extracción de información, eliminación de duplicados y ordenación

Mostrar por pantalla los diferentes *shells* de los usuarios del fichero *passwd*

```
manuel@x99:~$ cut -d ":" -f 7 passwd
/usr/bin/zsh
/usr/sbin/nologin
/usr/sbin/nologin
/usr/sbin/nologin
/bin/sync
...
```

Mostrar por pantalla los diferentes *shells* de los usuarios del fichero *passwd* ordenados alfabéticamente

```
manuel@x99:~$ cut -d ":" -f 7 passwd | sort
/bin/bash
/bin/false
/bin/false
/bin/false
/bin/false
/bin/false
/bin/sync
/usr/bin/zsh
/usr/bin/zsh
/usr/sbin/nologin
...
```

Mostrar por pantalla los diferentes *shells* de los usuarios del fichero *passwd* ordenados alfabéticamente sin duplicados

```
manuel@x99:~$ cut -d ":" -f 7 passwd | sort -u
/bin/bash
/bin/false
```

```
/bin/sync
/usr/bin/zsh
/usr/sbin/nologin
```

Di cuantos usuarios tienen cada uno de los tipos de *shell*

```
manuel@x99:~$ cut -d ":" -f 7 passwd | sort | uniq -c
      1 /bin/bash
      5 /bin/false
      1 /bin/sync
      2 /usr/bin/zsh
     45 /usr/sbin/nologin
```