Elias Ramirez – RamirezS4

Elias Ramirez – RamirezS4

Forencia en equipos Linux

Realizar un volcado de la memoria volatil (RAM) en sistemas Linux

Tenemos muchas opciones pero sobre todas recomiendo “LiME ~ Linux Memory Extractor”. Es una herramienta opensource que permite la adquisición de la memoria RAM de sistemas basados en Linux. LiME trabaja a nivel de kernel. La forma de hacerlo es mediante la insercción de un módulo en el kernel de Linux (insmod).

La prueba de uso sera ejecutada en la maquina REMnux la cual ejecuta ubuntu 20.04.6 LTS.

1. Clonamos el repositorio https://github.com/504ensicsLabs/LiME y extraemos el contenido.

2. vamos a la ruta “src” y ejecutamos “make(busca el archivo makefile y se encarga de ejecutar las ordenes contenidas en el mismo)”, a continuacion este nos genera un ejecutable.

3. Lo que continuaria es usar insmod con el ejecutable, seleccionar la ruta de salida y el formato de salida.

```
insmod lime-5.4.0-122-generic.ko "path=/home/remnux/Downloads/Dump/Dump01.lime format=lime"
```

Explicacion del comando:

- El comando insmod se utiliza en sistemas basados en Linux para insertar (cargar) un módulo del kernel en el sistema en tiempo de ejecución.
- lime-5.4.0-122-generic.ko Archivo generado luego de usar make.
- path=/home/remnux/Downloads/Dump/Dump01.lime: ruta de salida, y nombre del volcado.
- format=lime: formato de salida.

4. Por ultimo podemos usar por ejemplo volatily2 para saber el perfil del volcado recien generado:

```
vol.py -f /home/remnux/Downloads/Dump/Dump01.lime imageinfo
```

Elias Ramirez – RamirezS4

Rutas de los logs

Los logs (registros) en sistemas Linux son archivos que contienen información detallada sobre eventos y actividades del sistema. Estos registros son esenciales para el monitoreo, el diagnóstico de problemas, la seguridad y la auditoría en entornos Linux.

Ubicación del log	Datos registrados en el log y funcionalidad
/var/log/auth.log	Proporciona un registro de todas las actividades que implican un proceso de autenticación. Por ejemplo registra los usuarios logueados al sistema operativo. Registra el día, hora, usuario y ordenes que se han ejecutado con el comando sudo, los cronjobs que se han ejecutado, los intentos fallidos de autenticación, etc.
/var/log/debug	Para registrar datos de los programas que están actuando en modo depuración. De esta forma los programadores pueden obtener información si sus programas están funcionando adecuadamente.
/var/log/syslog	Contiene la totalidad de logs capturados por syslog. Por lo tanto en este fichero encontraremos multitud logs y será difícil de consultar y filtrar. Por este motivo, los logs se distribuyen en otros ficheros siguiendo la configuración del fichero /etc/rsyslog.conf.
/var/log/dmesg	Dentro del fichero encontraremos información relacionada con el hardware de nuestro equipo. Por lo tanto podremos obtener información para concluir si nuestro hardware funciona de forma adecuada.
/var/log/apache2/access.log	cada línea representa una solicitud HTTP a tu servidor web y proporciona información sobre la solicitud

/var/log/apache2/error.log	registra mensajes de error y advertencias relacionadas con el servidor web tambien con solicitudes que no se pudieron procesar
/var/log/nginx/access.log	Similar a los logs de acceso de apache
var/log/nginx/error.log	Proporciona información sobre errores y problemas específicos del servidor, como problemas con las solicitudes, entre otros.
/var/log/messages	Contiene mensajes informativos y no críticos de la actividad del sistema operativo. Acostumbra a contener los errores que se registran en el arranque del sistema que no estén relacionados con el Kernel. Por lo tanto, si no se inicia un servicio, como por ejemplo el servidor de sonido, podemos buscar información dentro de este archivo.
/var/log/faillog	Registra los intentos fallidos de autenticación de cada usuario. Dentro del archivo se almacena una lista de usuarios, los fallos totales de cada usuario, el número de fallo máximos que permitimos y la fecha y hora del último fallo. Si un usuario supera el número de fallos máximos establecidos se deshabilitará el usuario por el tiempo que nosotros fijemos.
/var/log/btmp	Almacena los intentos fallidos de logins en un equipo. Si alguien realizará un ataque de fuerza bruta a un servidor ssh, el fichero registraría la IP del atacante, el día y hora en que ha fallado el login, el nombre de usuario con que se ha intentado loguear, etc.
/var/log/lastlog	Ayuda a ver la fecha y la hora en que cada usuario se ha conectado por última vez.

Algunas otras rutas a destacar:

/var/log/wtmp	En todo momento contiene los usuarios que están logueados al sistema operativo.
/var/log/boot.log	Información relacionada con el arranque del sistema. Podemos consultarlo para analizar si se levantan los servicios del sistema, si se levanta la red, si se montan las unidades de almacenamiento, para averiguar un problema que hace que nuestro equipo no inicie, etc.
/var/log/cron	Registra la totalidad de información de las tareas realizadas por cron. Si tienen problemas con la ejecución de tareas tienen que consultar este log para ver si el trabajo se ha ejecutado o da errores. Debian no dispone de este log, pero encontrarán la misma información en /var/log/syslog. En Debian pueden configurar el fichero de configuración /etc/rsyslog.conf para generar un log especifico para cron.
/var/log/daemon.log	Registra la actividad de los demonios o programas que corren en segundo plano. Para ver si un demonio se levanto o está dando errores podemos consultar este log. Dentro de daemon.log encontraremos información sobre el demonio que inicia el gestor de inicio, el demonio que inicia la base de datos de MySQL, etc.
/var/log/dpkg.log	Contiene información sobre la totalidad de paquetes instalados y desinstalados mediante el comando dpkg.
/var/log/apt/history.log	Detalle de los paquetes instalados, desinstalados o actualizados mediante el gestor de paquetes apt-get.
/var/log/apt/term.log	Contiene la totalidad de información mostrada en la terminal en el momento de instalar, actualizar o desinstalar un paquete con apt-get.
/var/log/alternatives.log	Registra todas las operaciones relacionadas con el sistema de alternativas. Por lo tanto, todas las acciones que realicemos usando el comando update-alternatives se registrarán en este log. El sistema de alternativas permite definir nuestro editor de texto predeterminado, el entorno de escritorio predeterminado, la versión de java que queremos usar por defecto, etc.
/var/run/utmp	Ver los usuarios que actualmente están logueados en un equipo.

Elias Ramirez – RamirezS4

Busquedas iniciales

Buscar cambios de grupos

Puede enumerar los procesos de grupo buscando las palabras "groupadd" y "usermod" en el archivo auth.log. Enumerar los cambios de grupo en el rango de fechas del ataque facilitará el seguimiento de las acciones realizadas por el atacante.

grep “groupadd” /var/log/auth.log

grep “usermod” /var/log/auth.log

Identificar usuarios que han iniciado sesión en el sistema

El archivo /var/log/auth.log se puede examinar para detectar usuarios que iniciaron sesión en el sistema a través de SSH. Este archivo incluye inicios de sesión exitosos, así como inicios de sesión fallidos. De esta manera, podemos detectar ataques de fuerza bruta desde el archivo auth.log.

Elias Ramirez – RamirezS4

Identificación de usuarios que pueden realizar SSH

Puede obtener información sobre los usuarios que pueden realizar RDP en sistemas operativos Windows enumerando los usuarios incluidos en el grupo "Usuarios de escritorio remoto". Sin embargo, no existe un grupo similar en Linux. Los usuarios con permisos SSH se detectan en /etc/ssh/sshd_config . Si se especifica "AllowUsers" en este archivo, significa que otros usuarios no pueden utilizar el servicio SSH.

Busqueda de archivos sospechosos

Extensiones de archivos sospechosas

Con la ayuda del siguiente comando de búsqueda, podemos identificar los archivos con extensiones .sh, .php, .php7 y .elf en el sistema de archivos.

```
find / -type f \( -iname \*.php -o -iname \*.php7 -o -iname \*.sh -o -iname \*.elf \)
```

Busqueda por hora de modificacion

Podemos buscar archivos dentro del sistema de archivos según su tiempo de modificación.

podemos enumerar los archivos debajo del directorio /tmp que se han modificado entre las fechas del 25/01/2024 00:00:00 y el 25/01/2024 23:59: 00.

find /tmp -newermt "2024-01-25 00:00:00" ! -newermt "2024-1-25 23:59:00"

Busqueda por usuario propietario

Mientras buscamos archivos sospechosos, si conocemos a los usuarios comprometidos, realizar un análisis de los archivos propiedad de los usuarios comprometidos puede ayudarle a obtener resultados rápidos.

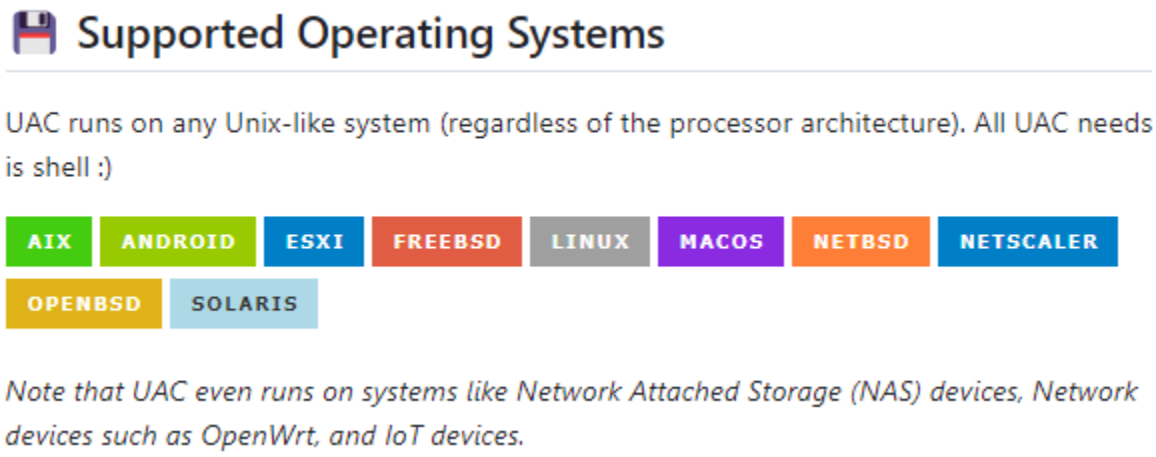
find /tmp -user www-data

UAC (Unix-like Artifacts Collector)

Es una herramienta muy potente, realiza un trabajo similar al hecho por Kape pero esta funciona en equipos Unix-like (Linux, BSD, Solaris, macOS, entre otros).

Según la descripcion dada por los mismos desarroladores: “UAC es un script de recopilación de Live Response para Incident Response que utiliza herramientas y binarios nativos para automatizar la recopilación de artefactos de los sistemas AIX, Android, ESXi, FreeBSD, Linux, macOS, NetBSD, NetScaler, OpenBSD y Solaris. Fue creado para facilitar y acelerar la recopilación de datos y depender menos del soporte remoto durante las tareas de respuesta a incidentes”. Se puede obtener desde su repositorio oficial https://github.com/tclahr/uac?tab=readme-ov-file.

Requisitos



Uso

Los escenarios de uso comunes pueden incluir los siguientes:

Recopile todos los artefactos según el `full` perfil y cree el archivo de salida en formato `/tmp`.

```
./uac -p full /tmp
```

Recopile todo `live_response` y el `bodyfile/bodyfile.yaml` artefacto y cree el archivo de salida en el directorio actual.

```
./uac -a live_response/\*,bodyfile/bodyfile.yaml .
```

Recopile todos los artefactos según el `full` perfil, pero excluya el `bodyfile/bodyfile.yaml` artefacto y cree el archivo de salida en formato `/tmp`.

```
./uac -p full -a \!bodyfile/bodyfile.yaml /tmp
```

Recopile el volcado de memoria y luego todos los artefactos según el `full` perfil.

```
./uac -a artifacts/memory_dump/avml.yaml -p full /tmp
```

Recopile el volcado de memoria y luego todos los artefactos según el `ir_triage` perfil, excluyendo el `bodyfile/bodyfile.yaml` artefacto.

```
./uac -a ./artifacts/memory_dump/avml.yaml -p ir_triage -a \!artifacts/bodyfile/bodyfile.yaml /tmp
```

Recopile todos los artefactos según el `full` perfil, pero limite la recopilación de datos según el intervalo de fechas proporcionado.

```
./uac -p full /tmp --date-range-start 2021-05-01 --date-range-end 2021-08-31
```

Recopile todos los artefactos excepto la respuesta en vivo de una imagen de disco de Linux montada en `/mnt/ewf`.

```
./uac -p full -a \!live_response/\* /tmp --mount-point /mnt/ewf --operating-system linux
```

Para mas informacion ir al enlace https://github.com/tclahr/uac?tab=readme-ov-file#-using-uac

Elias Ramirez – RamirezS4

Caso practico

De forma reciente realice un desafio en la plataforma lets defender en la cual debemos realizar una investigacion de los artefactos recopilados con esta herramienta, este desafio practico brinda informacion sobre todo lo que se puede recopilar con esta herramienta al mismo tiempo que muestra como realizar una investigacion.

https://github.com/ramirezs4/Tips-and-tools-forensics---RS4/blob/main/Letsdefend-Confluence-CVE-2023-22527.pdf