PAI 1. SISTEMA DE DETECCIÓN DE INTRUSOS PARA HOST BASADO EN VERIFICADORES DE INTEGRIDAD

Escuela Técnica Superior de
Ingeniería Informática

Índice

Alcance y Objetivos del Proyecto

En este proyecto, se nos ha introducido a un nuevo tipo de servicio a implementar en una organización. Este se trata de un sistema de detección de intrusos para host, HIDS, basado en un verificador de integridad, donde se nos ha indicado en la Política de Seguridad que "debe verificarse diariamente la integridad de los ficheros binarios/directorios de los sistemas informáticos críticos y las aplicaciones/páginas Web de la organización y dar cuenta mensualmente al ISG de la organización de los resultados diarios de la verificación".

Por tanto los objetivos en este proyecto son investigar y escoger el HIDS más conveniente de acuerdo a la Política de Seguridad, desplegar un servicio con verificaciones en intervalos mínimos de una hora y almacenar los informes para generar informes con periodicidad, establecimiento de un sistema de alertas y monitorización y un informe con la realización de las pruebas.

Recursos para la Auditoría

Recursos Humanos

El equipo de seguridad para esta asignatura se compone de los siguientes miembros:

- Barragán Candel, Marina - estudiante de Ing. Informática – Tecnología Informática, en la mención de Tecnologías de la información
- Calcedo Vázquez, Ignacio - estudiante de Ing. Informática – Tecnología Informática, en la mención de Sistemas de Información
- Polo Domínguez, Jorge - estudiante de Ing. Informática – Ingeniería de Computadores
- Sala Mascort, Jaime Emilio - estudiante de Ing. Informática – Tecnología Informática, en la mención de Computación

El equipo se organizará mediante la herramienta de Github, bajo el repositorio público SSII.

Estudio Inicial

En primera instancia, debemos investigar sobre los HIDS, viendo sus funciones básicas y ventajas y debilidades a tener en cuenta para realizar la implementación más adecuada.

Un HIDS es un servicio que implementa la comprobación de la integridad de los datos, mediante mecanismos de generación y comparación de hashes en el propio host, teniendo acceso privilegiado a información del sistema a proteger.

Una de las grandes ventajas es que detectan fácilmente ataques dentro del equipo, ya que suelen monitorizar inicios de sesión, cambios en ficheros y registros. Además, requieren menos recursos que otros sistemas como los Network Intrusion Deteccion Systems, NIDS, que pueden afectar al rendimiento del sistema.

Sin embargo, estos sistemas tienen una respuesta más lenta, ya que detectan los ataques una vez que ya se han producido, haciendo que sea vital que los informes del servicio no puedan ser borrados ni modificados, además de evitar la interrupción del servicio.

Otro factor a tener en cuenta es la colisión de hashes, que se produce cuando la función de generación de hashes no es la adecuada, por lo que se debe equilibrar la facilidad de generación con la dificultad para la colisión.

Estudio de las Soluciones Disponibles

Hoy en día, hay multitud de opciones a la hora de implementar esta herramienta, a continuación se van a poner algunas de ellas con sus características principales y precios.

Tripwire

Tripwire es un programa que proporciona una herramienta de seguridad e integridad de datos, desarrollado de forma Open Source.

Sus principales características son las siguientes:

- Servicio de monitorización y alerta de cambios en un sistema de ficheros;
- Comparación de la firma digital de directorios y archivos frente a una base de datos de los mismos;
- Generación de la base de datos que contendrá una copia de la información necesaria para comprobar la integridad de los archivos, accesible mediante contraseña cifrada. Esta base de datos no puede tener un efecto con carácter retroactivo, si el sistema ya había sido modificado previamente.

Wazuh

Wazuh es una plataforma de código libre y gratuito para la detección de intrusos, monitorización de la seguridad y servicio de alertas y respuestas ante intrusiones. Se puede hacer uso en sistemas finales, servidores en la nube y contenedores. Además permite la recolección y análisis de datos mediante herramientas externas.

Sus principales características son las que siguen:

- Informes de seguridad, se recoge, indexa y analiza información de distintas organizaciones colaboradoras para un proceso de mejora continua en los sistemas.
- Detección de intrusos, el cliente de Wazuh escanea los sistemas monitorizados en busca de anomalías.
- Monitorización de la integridad de ficheros, se trata de una monitorización de los archivos del sistema en busca de cambios de permisos, atributos y propiedad de los mismos, además de identificar que usuarios o aplicaciones han realizado los cambios.
- Servicio de Alerta y Actuación ante incidentes, de forma nativa, Wazuh provee de acciones predefinidas ante amenazas cuando se cumplen ciertos criterios.

Solución Propia

Finalmente, se ha optado por afrontar el proyecto desarrollando un HIDS propio para poder obtener una mejor comprensión de los conceptos y como integrar servicios en los sistemas Linux, que dominan el mercado de los servidores, haciendo del aprendizaje de los mismos una gran ventaja en el mercado laboral.

Este servicio contará con las siguientes características:

- Monitorización de la integridad de ficheros, esta se realizará mediante la comprobación y comparación de hashes entre los distintos estados temporales de los mismos.
- Monitorización del árbol de directorios, comprobación de la estructura de directorios
- Servicio de Alerta ante incidencias, se integrará un sistema de alertas por correo mediante los paquetes de mailutils de Linux, aunque se ha descartado la implementación del servidor smtp necesario, debido a que no entra de los ámbitos de la asignatura,

optando por correos locales al root del sistema, más información en el Anexo I: Manual de Configuración.

- Informes de seguridad, se irá generando un registro del porcentaje de cambios entre cada ciclo del servicio.

Actividades del Proyecto

Las actividades que se han desarrollado en el proyecto han sido las siguientes:

Investigación de los Servicios HIDS

En primera instancia, cada miembro del equipo investigó sobre estos servicios y las herramientas comerciales que los implementan con el fin de tener una reunión donde se haya llevado un estudio previo para que todos los integrantes tuvieran conocimientos básicos a la hora de repartir tareas.

En el reparto de tareas, se formaron dos grupos uno que trató las herramientas comerciales, siendo este Jorge Polo y Marina Barragán. Y el otro grupo, formado por Ignacio Calcedo y Jaime Emilio Sala, que investigaron las funciones objetivo del proyecto y como y en que lenguaje se podrían implementar.

Finalmente, tras la investigación de las herramientas, se hicieron una serie de pruebas con las herramientas, pero se optó por el desarrollo del servicio.

Desarrollo de la Herramienta

Tras el reparto de tareas, el grupo de desarrollo se puso rápidamente de acuerdo en desarrollar un script en bash debido a las facilidades que tiene para integrarlo como un servicio en un sistema Linux. Por tanto, se procedió a su desarrollo con las funcionalidades básicas, en primer lugar manual, y posteriormente investigar systemd para integrarlo como un servicio del sistema.

Al finalizar el proceso de desarrollo, se decidió por llamar a la herramienta apicultor por parte de Ignacio Calcedo, exponiendo que existe cierta similitud entre los procedimientos de los apicultores y los HIDS, pues estos deben encargarse de vigilar y controlar el correcto desarrollo de los panales, y de igual manera deben vigilar los HIDS el conjunto de ficheros para que la integridad de los mismos se mantenga. Adicionalmente, se tomó inspiración del termino honeypot que se utiliza en otros tipos de software de seguridad informática

Desarrollo de Pruebas sobre el HIDS

Una vez quedó desarrollado el servicio , apicultor.service, era necesario desarrollar una serie de pruebas para garantizar el correcto funcionamiento del software diseñado por el equipo de desarrollo.

El fundamento de los HIDS reside en la generación de resúmenes los documentos de un directorio o conjunto de directorios y posteriormente realizar una comprobación de diferencias entre distintas generaciones de hashes para comprobar si en efecto se ha producido algún cambio y, por tanto, se han alterado los resúmenes con respecto a la versión previa. Partiendo de aquí, la forma más simple de comprobar que todo funcione es realizar cambios en una parte considerable de los ficheros disponibles en el sistema durante el período entre actualizaciones y comprobaciones. Esto nos informaría de manera inmediata si se producían modificaciones en los hashes y, lo más importante, si se notifica al usuario del sistema de que se han producido dichos cambios.

Resultado de las Pruebas

Funcionamiento del Servicios

En función de las pruebas realizadas, hemos comprobado que el servicio realiza las funciones requeridas por el cliente, donde se lleva un registro de los ciclos, hay un sistema de alertas, y se hace una comprobación periódica de los archivos y directorios.

Robustez del Servicio

Todos los sistemas HIDS suelen presentar una debilidad en común. Cualquier vulneración de la integridad de la base de datos/ficheros encargados de controlar que los hashes de los ficheros se mantienen intacto provoca que su efectividad se vea reducida drásticamente. Partiendo de aquí, es necesario tomar ciertas precauciones. Lo más importante es proteger el acceso a la carpeta /root: Tal y como está configurado el software Apicultor, la carpeta donde se almacena toda la información esencial del servicio reside en /root/.apicultor. Cualquier intento de entrar en ese directorio sin los permisos adecuados debe ser reflejado inmediatamente.

Sistema de Alerta

En cuanto al sistema de alertas, no se puede detectar un ataque a priori ya que dependemos directamente de la comprobación de ficheros por tanto es un límite al que no podemos hacer frente. Sin embargo, se podría mejorar el sistema apuntando que o quien ha sido el autor de una modificación.

A pesar de todo, se hace un aviso, tan pronto como sean los ciclos del servicio, dando información al administrador de los cambios que se han llevado a cabo.

Anexo I: Manual de Configuración del Servicio

Requisitos previos

Sistema Operativo Linux Like que habilite systemd por defecto. Esto incluye Sistemas Operativos como Debian GNU/Linux, Fedora, Ubuntu, Frugalware, Mageia, Mandriva, openSUSE, Archlinux o RedHat entre otros.

Método de Instalación

Para la instalación del servicio Apicultor.service, es necesario disponer de los siguientes ficheros:

- install.sh
- uninstall.sh

Estos dos ficheros son scripts en bash que se encargan de los procesos de instalación y desinstalación del servicio. Para el correcto uso de los mismos, es necesario modificar los permisos de ambos scirpts para que sean ejecutables. Usted tendrá que abrir una terminal y ejecutar el siguiente comando:

```
sudo chmod 700 /path/to/install.sh /path/to/uninstall.sh
```

Donde /path/to/ se refiere a la ruta donde se ubiquen los ficheros ya mencionados. En caso de que con la consola se ubique dentro del directorio que contiene ambos ficheros, con chmod 700 instalation.sh uninstall.sh será suficiente.

Una vez que haya conseguido configurar los permisos correctamente, el siguiente paso a seguir es comenzar la instalación. Los desarrolladores recomiendan ejecutar primer el script de desinstalación para asegurar un funcionamiento correcto y limpio. Para ello, ejecute el siguiente comando:

```
sudo ./path/to/uninstall.sh
```

Si no ha utilizado este programa antes, es casi seguro que obtendrá la siguiente salida:

"Procediendo a la eliminación de archivos y directorios de apicultor y su servicio"

Esto es buena señal, no es necesario alarmarse por la salida de error. Es momento de pasar al siguiente paso, la instalación del servicio. Para ello, tendrá que ejecutar en la consola el siguiente comando:

```
sudo ./path/to/install.sh [--timer=XX] /path/to/watch
```

Vamos a desglosar este comando. Siguiendo el standard explicado previamente, path/to/install marca la ruta del fichero ejecutable de instalación. Los parámetros siguientes son sin embargo más cruciales. --timer=XX indica la frecuencia en segundos con la que se revisará que se han producido cambios en los resúmenes (hashes) de los ficheros, es decir, que se hayan alterado, ya sea de manera ajena o determinada por el usuario. En caso de no incluir este parámetro en la ejecución, se configurará por defecto a 3600 segundos, es decir, una hora de intervalo entre comprobación y comprobación. Por último, el parámetro /path/to/watch describe la ruta del directorio en el que el servicio deberá enfocar su recolección de hashes. Sin este parámetro,el script le mostrará una pantalla de ayuda con parámetros extra del servicio, pero no se instalará.

Una vez que haya introducido todos los parámetros deseados, deje que termine de su sistema operativo se encargue de instalar todos los paquetes necesarios para que pueda disfrutar del servicio. Si no se tenían previamente todos los paquetes necesarios, nos aparecerá "¿Desea continuar? [S/n]" para realizar la instalación de los paquetes, introduzca una S.

Una vez que termine la instalación, el sistema le mostrará el estado del servicio y algunos datos adicionales por pantalla, dando por concluida la instalación. Para comprobar que el servicio está activo, introduzca el siguiente comando por consola:

```
sudo systemctl status apicultor.service | more
```

Este comando le permitirá comprobar si el servicio está operativo en el sistema.

Instalación de Mailutils

Una parte esencial del servicio es la notificación mediante un servicio de correo local si se han producido cambios en los ficheros a revisar en el sistema. Para ello, el servicio Apicultor.service se basa en la herramienta mailutils para el envío de correos. Por defecto, si toda la instalaciónprevia ha ido correctamente, mailutils debería haberse instalado con éxito.

Sin embargo, para asegurar que todo funciona correctamente, le recomendamos que comrpuebe si tiene instalado dicho software. En caso de no poseerlo, procedemos a describirle cómo instalarlo.

En primer lugar, abra una consola de comandos. Una vez abierta, teclee el siguiente comando:

```
sudo apt-get install mailutils
```

De manera similar al servicio anterior, le mostrará los paquetes a instalar y le solicitará continuar con la instalación. Tras esto, el programa se instalará con éxito. Tras un breve proceso de instalación, le saldrá una pantalla con título Postfix Configuration. Aquí podrá elegir qué opción se ajusta más a sus necesiadades de correo. Las opciones que se le ofrecen para sets de preconfiguración del software son:

- Sin configuración
- Sitio de Internet
- Internet con <<smarthost>>
- Sistema satélite
- Sólo correo local

De todas las anteriores, solo debe elegir la opción "Sólo correo local". Desplace la barra de selección sobre ésta y pulse aceptar. Esto le llevará a otra ventana, esta vez para preguntar qué nombre desea asignar a su sistema de correo. Escriba cualquiera que le resulte conveniente. Por defecto le asignará el nombre de la máquina. Una vez introducido el nombre, pulse aceptar y espere a que el sistema termine de instalar el servicio.

Una vez terminada la instalación, ya podrá disfrutar de su sistema de correo local, al que podrá acceder con sudo mail.mailutils.

Anexo II: Pruebas Realizadas

Tras la instalación del servicio, nos disponemos a enumerar las pruebas realizadas.

Funcionamiento del Servicio

Mediante el comando `systemctl status apicultor.service`, podemos comprobar como el servicio está cargado y a la espera de que se cumplan las condiciones de disparo automático como figura en el output de consola inferior.

```
● apicultor.service - HIDS service that manages the integrity of files and directories in the given paths
     Loaded: loaded (/etc/systemd/system/apicultor.service; disabled; vendor preset: enabled)
     Active: activating (auto-restart) since Sun 2020-11-01 12:47:06 CET; 148ms ago
    Process: 6687 ExecStart=/usr/bin/apicultor /etc/ (code=exited, status=0/SUCCESS)
   Main PID: 6687 (code=exited, status=0/SUCCESS)
nov 01 12:47:06 ssii-VirtualBox systemd[1]: apicultor.service: Succeeded.
nov 01 12:47:06 ssii-VirtualBox systemd[1]: apicultor.service: Scheduled restart job, restart counter is at 2.
nov 01 12:47:06 ssii-VirtualBox systemd[1]: Stopped HIDS service that manages the integrity of files and
directories in the given paths.
nov 01 12:47:06 ssii-VirtualBox systemd[1]: Started HIDS service that manages the integrity of files and
directories in the given paths
```

Además, podemos también determinar que está funcionando correctamente, ya que se están generando los ficheros con los históricos de hashes, mediante el comando `sudo ls -la /root/.apicultor/history`, que nos ofrece el siguiente output.

```
2020-11-01_12:47_directorios.csv  2020-11-01_13:02_directorios.csv
2020-11-01_12:47_hashes.csv       2020-11-01_13:02_hashes.csv
2020-11-01_12:52_directorios.csv  2020-11-01_13:07_directorios.csv
2020-11-01_12:52_hashes.csv       2020-11-01_13:07_hashes.csv
2020-11-01_12:57_directorios.csv  2020-11-01_13:12_directorios.csv
2020-11-01_12:57_hashes.csv       2020-11-01_13:12_hashes.csv
```

Tras más de 20 minutos de prueba, podemos comprobar que el servicio devuelve correctamente cada 5 minutos (pues se ha configurado el parámetro de temporizador a 300 segundos) un csv con los directorios del sistema y otro con los hashes de los distintos ficheros que se encuentran en el mismo.

Sistema de Alertas

Para realizar estas pruebas se ha establecido el siguiente método. Mostraremos la hora del sistema, empezaremos a modificar ficheros de una carpeta en la que tendremos acceso a varios,mostraremos los cambios realizados con Git para dar constancia de esto y posteriormente comprobaremos que todo ha ido en orden abriendo mail.mailutils y comprobando el mensaje más reciente.

```
ssii@ssii-VirtualBox:~/Documentos/SSII/PAI1$ w
 14:12:41 up  1:34,  1 user,  load average: 0,00, 0,03, 0,05
USUARIO  TTY      DE               LOGIN@   IDLE   JCPU   PCPU WHAT
ssii     :0       :0               12:38   ?xdm?   3:20   0.01s /usr/lib/gdm3


ssii@ssii-VirtualBox:~/Documentos/SSII/PAI1$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios no rastreados para el commit:
  (usa "git add <archivo>..." para actualizar lo que será confirmado)
  (usa "git restore <archivo>..." para descartar los cambios en el directorio de trabajo)
        modificado:     Objetivo 1.txt
        modificado:     Objetivo 2.txt
        modificado:     Objetivo 3.txt
        modificado:     informe_apicultor.txt
```

```
"/var/mail/root": 5 mensajes 5 nuevos
>N   1 root               dom nov  1 13:52  32/1266  Cambios en ficheros de
 N   2 root               dom nov  1 13:57  39/1821  Cambios en ficheros de
 N   3 root               dom nov  1 14:02  37/1647  Cambios en ficheros de
 N   4 root               dom nov  1 14:07  32/1270  Cambios en ficheros de
 N   5 root               dom nov  1 14:12  32/1266  Cambios en ficheros de

Message-Id: <20201101131228.072E521C4D@ssii-VirtualBox.home>
Date: Sun,  1 Nov 2020 14:12:28 +0100 (CET)
From: root <root@ssii-VirtualBox>

--639821885-1604236347=:18389
Content-Type: text/plain; charset=UTF-8
Content-Disposition: attachment
Content-Transfer-Encoding: 8bit
Content-ID: <20201101141227.18389.1@ssii-VirtualBox>

Ha habido cambios en los archivos adjuntos

--639821885-1604236347=:18389
Content-Type: application/octet-stream; name="diffHashes.csv"
Content-Disposition: attachment; filename="/root/.apicultor/diffHashes.csv"
Content-Transfer-Encoding: base64
Content-ID: <20201101141227.18389.1@ssii-VirtualBox>

aW5mb3JtZV9hcGljdWx0b3IudHh0LDBiMzZlYmM2YThlY2YzZmNhYzE5Yjg3ODY2YWFmZjBkODFl
NDlmYjdlMTc4YWY1YTUyZjRiN2RiZjJlMDNhNzAsL2hvbWUvc3NpaS9Eb2N1bWVudG9zL1NTUkv
UEFJMS9pbmZvcm1lX2FwaWN1bHRvci5OeHQsTU9ESUZJRUQK
--639821885-1604236347=:18389--
```

Como se puede comprobar, el sistema notifica con éxito los cambios realizados pues el fichero que comprueba que se ha producido una variación entre los hashes.