|  |
| --- |
|  |

|  |
| --- |
| Javier Hidalgo García  Juan Pedro Hurtado Masero  Nicolás Sibello Litrán |

SECURITY TEAM 11

|  |  |
| --- | --- |
|  | SISTEMA DE DETECCIÓN DE INTRUSOS (HIDS) PARA ALMACENAMIENTO MASIVO BASADO EN VERIFICADORES DE INTEGRIDAD |



SEGURIDAD EN SISTEMAS INFORMÁTICOS E INTERNET

PAI-1 HIDS

contenido

Escribir el título del capítulo (nivel 1)1

Escribir el título del capítulo (nivel 2)2

Escribir el título del capítulo (nivel 3)3

Escribir el título del capítulo (nivel 1)4

Escribir el título del capítulo (nivel 2)5

Escribir el título del capítulo (nivel 3)6

|  |
| --- |
| INTRODUCCIÓN |

|  |  |  |
| --- | --- | --- |
| Problema planteado Un enfoque clásico de la seguridad de un sistema informático siempre define como principal defensa del mismo sus controles de acceso (desde una política implantada en un cortafuegos hasta unas listas de control de acceso en un router o en el propio sistema de ficheros de una máquina), en muchos casos, esos controles no pueden protegernos ante un ataque. Desde un pirata informático externo a nuestra organización a un usuario autorizado que intenta obtener privilegios que no le corresponden en un sistema, nuestro entorno de trabajo no va a estar nunca a salvo de **intrusiones**.  Llamaremos intrusión a un conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso; analizando esta definición, podemos darnos cuenta de que una intrusión no tiene por qué consistir en un acceso no autorizado a una máquina: también puede ser una negación de servicio. A los sistemas utilizados para detectar las intrusiones o los intentos de intrusión se les denomina **sistemas de detección de intrusiones** (Intrusion Detection Systems, IDS), cualquier mecanismo de seguridad con este propósito puede ser considerado un IDS, pero generalmente sólo se aplica esta denominación a los sistemas automáticos (software o hardware): es decir, lo habitual (y lógico) es que a la hora de hablar de IDSes no se contemplen el resto de casos.  Cualquier sistema de detección de intrusos ha de cumplir algunas propiedades para poder desarrollar su trabajo correctamente: el IDS ha de ejecutarse continuamente sin nadie que esté obligado a supervisarlo. Otra propiedad, y también como una característica a tener siempre en cuenta, es la *aceptabilidad* o grado de aceptación del IDS; al igual que sucedía con cualquier modelo de autenticación, los mecanismos de detección de intrusos han de ser aceptables para las personas que trabajan habitualmente en el entorno. Por ejemplo, no ha de introducir una sobrecarga considerable en el sistema (si un IDS ralentiza demasiado una máquina) ni generar una cantidad elevada de falsos positivos (detección de intrusiones que realmente no lo son) o de logs, ya que entonces llegará un momento en que nadie se preocupe de comprobar las alertas emitidas por el detector.  Una tercera característica a evaluar a la hora de hablar de sistemas de detección de intrusos es la *adaptabilidad* del mismo a cambios en el entorno de trabajo. Como todos sabemos, ningún sistema informático puede considerarse estático: todo cambia con una periodicidad más o menos elevada. Si nuestros mecanismos de detección de intrusos no son capaces de adaptarse rápidamente a esos cambios, están condenados al fracaso.  Un sistema de detección de intrusos basado en máquina (host-based IDS o HIDS) es un mecanismo que permite detectar ataques o intrusiones contra la máquina sobre la que se ejecuta. El IDS basado en host o **HIDS** monitorea las características de un host y los eventos que ocurren en él en busca de actividades maliciosas o sospechosas. Un host es un equipo o dispositivo conectado a la red. El HIDS puede identificar tanto el tráfico malicioso que entra en el host como el que origina en el propio host y que un sistema de detección basado en red no podría detectar. En definitiva, un HIDS engloba:   * Detectar y detener los ataques directos potenciales, pero no realizar un análisis en busca de malware. * Identificar ataques potenciales y enviar alertas, pero no detener el tráfico. * Combinar las funcionalidades de aplicaciones antimalware con protección de firewall. * Es un sistema sin agente que analiza los archivos en un host en busca de posible malware.  Políticas y controles de seguridad La política a seguir en la empresa es la siguiente:  “Debe verificarse diariamente la integridad de los ficheros binarios, de imágenes y directorios de los sistemas informáticos críticos y las aplicaciones de la organización y dar cuenta mensualmente al ISG de la organización de los resultados diarios de la verificación” Solución propuesta DESARROLLO E IMPLEMENTACIÓN  **Script Phyton: HIDS.py**  Hemos decidido implementar un modelo basado en árbol binario:  De esta manera ahorramos muchísimo tiempo para la búsqueda de un archivo que ha sido atacado, la clase árbol binario contiene varias funciones:     * **getFileHash** (path, buffer\_size): esta función obtiene el hash de un archivo, utilizando el tamaño de un buffer de 64Kb. Se obtiene el hash mediante el algoritmo *sha256*.      * **getAllFilesInDirectory** (mainPath, files): devuelve una lista con todas las rutas de los archivos a proteger. * **createBST**(ids, files, a, b): función que crea el árbol binario completamente balanceado a partir de una lista ordenada con todos los *ids* y rutas de ubicación de los ficheros.      * **searchFileById** (root,Id): función que realiza la búsqueda binaria en el árbol generado. * **checkIntegrity** (tree,ids): esta función comprueba si ha sido comprometida la integridad de los archivos que se le pasan. Para un fichero, primero realiza la búsqueda binaria en el árbol y calcula el *MAC (sha256)*. Después compara ambos *MAC* para, en caso de haber cambiado en el periodo estipulado, anotar en el *log* que el archivo ha sido comprometido.   La configuración del *log* se realiza con la sentencia *logging.basicConfig()*.    **Enviar\_email.py**  Poseemos también otro archivo *py* llamado **envia\_email.py**;este script nos permite enviar un correo a los empleados de la empresa informándoles del reporte de ataques a la integridad mensualmente en el directorio especificado mediante una función llamada **envia().**    **Params.py**  Este archivo py se encarga de traducir el archivo **PARAMETERS.conf** de manera que sea legible para todas las funciones que necesiten de estos parámetros, de esta manera nos queda un programa que tiene una parametrización muy sencilla y simple.     * **loadHids** () : traduce los diferentes parámetros de PARAMETERS.conf que se necesitan para hids.py      * **loadMail** () : traduce los diferentes parámetros de PARAMETERS.conf que se necesitan para parameters.py   **PARAMETERS.conf**  Archivo que guarda todos los parámetros que necesita nuestro programa:   * Directorio\_Base: Directorio a analizar de ataques * Tiempo en segundos : Tiempo escalado a 1 día (10 segundos) * Numero de comprobaciones : Cantidad de días para 1 mes (30) * Sender : Emisor del correo * Username : Usuario del emisor * Password : Contraseña del emisor * Server * Puerto * Destinatarios : Correos receptores * Asunto : Asunto del correo * Ruta del registro : Raíz donde se encuentra el registro.log que será enviado * Cuerpo : Cuerpo del mensaje  Prueba Los parámetros escalados a las pruebas que hemos realizado para ver que todo funciona correctamente son las siguientes:   * 10 segundos corresponde a 1 día y 5 minutos corresponde a 1 mes.   En primer lugar ponemos los parámetros correspondientes:    Corremos el programa y nos crea un árbol binario de 1000 archivos en 35 segundos | | |
| Ahora vamos a modificar un archivo de ese directorio para comprobar si detecta ataques a la integridad y lo registra en el archivo log que será enviado. | | |
| Como podemos ver, el registro guarda correctamente la fecha y nombre del archivo que se ha modificado  Ahora si revisamos el buzón de los correos que hemos puesto como destinatarios podemos observar como se ha recibido el archivo .log con esta misma entrada | | |
|  | | |
|  |  |  |

|  |
| --- |
|  |

|  |
| --- |
| BIBLIOGRAFÍA |

Sistemas de detección de intrusos (Seguridad en Unix y Redes) – Red IRIS

El sistema de detección de intrusiones (IDS) – Grupo Ático 4 (ciberseguridad)

Banco de Preguntas y Respuestas de redes, Categoría: CCNA CyberOps - ¿Qué es un sistema de detección de intrusiones basado en host (HIDS)?