Apuntes de Ciberseguridad: Métodos y Herramientas para la Protección de Servidores y Sistemas Operativos Open Source

Ruben Apablaza Muñoz
-OZonE-

TABLA DE CONTENIDOS

10. Comandos y Herramientas de Seguridad para Hardening en SO Open Source – Seguridad de Red
    - Herramientas Iptables y Firewalld
        - iptables
        - firewalld
        - Principios Teóricos Clave
11. Registro de Auditoría del Sistema
12. Gestión de Paquetes y Actualizaciones en Entornos Open Source
    - Gestor de Paquetes
    - Repositorios
    - Dependencias
    - Actualizaciones e Instalación
    - Seguridad
    - Rollback
    - Automatización
    - Distribuciones de software LIBRE v/s PROPIETARIO
13. Seguridad en la Gestión Remota de Sistemas y los Comandos Asociados a la Configuración de SSH
    - Conceptos Clave (Cifrado, Autenticación, Integridad de datos)
    - Configuración y Comandos Básicos
    - Mejores Prácticas de Seguridad para SSH
14. Seguridad en las Conexiones hacia el Sistema Operativo a Través de una Herramienta para Crear VPNs
    - Configuración de VPN utilizando una herramienta de código abierto
    - Autenticación y Cifrado de la Conexión VPN
    - Medidas de Seguridad Adicionales en la Conexión VPN
    - Tráfico de la Conexión VPN

METODOS Y HERRAMIENTAS PARA PROTEGER SERVIDOR Y SO OPEN SOURCE

- Protocolos de cifrado
  - Método matemático para codificar la información.
  - Entre los más conocidos están:
    - SSL/TLS
      - Seguridad web.
    - SSH
      - Conexiones seguras entre dispositivos finales.
    - PGP
      - Cifrado de correos electrónicos.

- Firewalls
  - Hardware o software o combinación de ambos.
  - Filtran el tráfico de red según un conjunto de reglas de seguridad.

- Autenticación de usuarios
  - Proceso mediante el que se comprueba la identidad de una persona para acceder a un sistema informático.
  - Métodos comunes:
    - Uso de contraseñas.
    - Tarjetas de identificación.
    - Reconocimiento biométrico.
    - Autenticación de 2 factores.

Antivirus

- Software que detecta y elimina malware.

- Control de acceso
  - Sistema que limita el acceso NO autorizado a los recursos.
  - En Linux los modelos de control de acceso son:
    - DAC (Control de Acceso Discrecional)
      - Método tradicional, el propietario de un recurso es quien decide quien puede acceder y a qué.
      - Los permisos se dividen en 3 categorías:
        - Propietario
        - Grupo
        - Otros
      - Los permisos son
        - Lectura (R)
        - Escritura (W)
        - Ejecución (X)

USUARIO Y PERMISOS	ls	muestra los archivos en un directorio.
USUARIO Y PERMISOS	ls /ld /home/rgui/Escritorio/Carpeta_Prueba	Consulta los permisos de la carpeta en la ruta señalada
USUARIO Y PERMISOS	chmod XXX /home/rgui/Escritorio/Carpeta_Prueba	Cambia los permisos al directorio en la ruta señalada, ojo no afecta a los archivos en su interior
USUARIO Y PERMISOS	chmod 777 hello.txt	da acceso total al archivo hello.txt al propietario del archivo, al grupo y a los usuarios
USUARIO Y PERMISOS	chown	Cambia el propietario de un archivo o directorio
USUARIO Y PERMISOS	chgrp	Cambia el grupo de un archivo o directorio

LISTAS DE CONTROL DE ACCESO Y PERMISOS ESPECIALES

- ACL (Listas de control de acceso)
  - Permiten especificar permisos detallados para usuarios y grupos individuales más allá de la estructura básica de permisos DAC. Esto quiere decir que se puede otorgar un acceso de escritura a un usuario especifico sin modificar los permisos generales o la pertenencia a grupos.
  - Incluyen permisos RWX.
  - Se gestionan mediante el comando “setfacl”.

`setfacl -m u:juan:rw archivo.txt`	LISTA DE ACCESO: Otorga al usuario "juan", permisos de lectura y escritura sobre el "archivo.txt" INDEPENDIENTE de los permisos DAC del archivo.

- Sticky Bit (Permiso especial)
  - Se aplica a DIRECTORIOS para restringir la eliminación y modificación de archivos quedando este privilegio solo a los propietarios del archivo, el propietario del directorio y el superusuario.
  - Muy útil en directorios compartidos como /tmp
  - Al aplicar Sticky Bit a un directorio, cualquier usuario puede crear archivos en él, pero solo el propietario del archivo y el superusuario pueden eliminarlo o renombrarlo.
  - Para usarlo se utiliza chmod +t

- setuid y setgid
    - Son esenciales para programas que requieren acceso a recursos protegidos

chmod u+s/path/to/program	PERMISOS ESPECIALES setuid. Permite que un programa se ejecute con los privilegios del propietario del archivo, en lugar de los del usuario que lo ejecuta.	setuid
chmod g+s/path/to/directory	PERMISOS ESPECIALES setgid. Similar al setuid pero el programa se ejecuta con los privilegios del grupo del propietario del archivo. Cuando se aplica a un directorio, los archivos creados dentro heredan el grupo del directorio.	setgid

REVISAR ¡!! Uso de “nosuid” en chat gpt y cuando conviene desactivar setuid

Control de acceso obligatorio (MAC)

- Se diferencia de DAC en que limita el acceso a los recursos del sistema según políticas de seguridad establecidas por los administradores del sistema *y no por los propietarios de los archivos*. Este enfoque basado en políticas asegura que el acceso a los recursos del sistema sea gestionado de manera mas rigurosa y conforme a las necesidades ESPECIFICAS de seguridad del entorno.
- En el contexto de SO Linux, SELinux y AppArmor son módulos de seguridad ejemplares que implementan MAC.
- SELinux
    - Desarrollado por la NSA.
    - Utiliza etiquetas de seguridad para asociar políticas a usuarios, procesos y objetos del sistema.
    - Puede operar en modos

        - Enfocado
        - Permisivo
    - Sus políticas se pueden personalizar.
- AppArmor
    - Enfoque basado en perfiles para los programas individuales.
    - Es mas sencillo de configurar que SELinux, se centra en la simplicidad y facilidad de uso.

La función esencial tanto de SELinux y AppArmor, es reforzar la seguridad al restringir las acciones que programas y procesos pueden realizar.

Su aplicación, ayuda a reforzar el sistema de:

- Protección contra malware: Al restringir lo que los programas pueden hacer, se reduce el riesgo de que el malware cause daño significativo.
- Minimización de daños: En caso de una brecha de seguridad, las políticas detalladas pueden limitar el impacto al restringir el acceso a recursos críticos.
- Seguridad en entornos de múltiples usuarios: Especialmente útil en servidores y sistemas donde múltiples usuarios tienen acceso a diferentes servicios.
- Cumplimiento de normativas: Facilita la adhesión a estándares de seguridad y regulaciones de la industria.

Comparativa dac y mac

- DAC (Discretionary Access Control)
  - Control Discrecional.
  - El dueño del recurso (archivo, carpeta) decide quién puede acceder y con qué permisos.
  - Es el modelo usado por defecto en Linux/Unix (chmod, chown, etc.).
  - Más flexible, pero menos seguro.

- MAC (Mandatory Access Control)
  - Control Obligatorio.
  - Las políticas de seguridad las define el sistema, no el usuario.
  - Ejemplo: SELinux, AppArmor.
  - Es más estricto y seguro, ideal para entornos críticos.

Resumen comparativo

Característica	DAC	MAC
¿Quién decide permisos?	El dueño del recurso	El sistema/administrador
Flexibilidad	Alta	Baja
Seguridad	Moderada	Alta
Uso común	Sistemas operativos estándar	Sistemas seguros o militares

CONTROL DE ACCESO BASADO EN ROLES (RBAC)

- Modelo que asigna permisos a ROLES en lugar de a USUARIOS.
- Los roles son creados en función de las tareas laborales.
- Simplifica la administración de permisos en entornos con muchos usuarios donde la administración de permisos individuales sería impracticable.
- Características:
  - Asignación de roles
  - Separación de deberes
  - Simplificación de la administración
  - Flexibilidad y escalabilidad
- Se aplica a entornos y plataformas, desde SO y BBDD hasta aplicaciones web y redes empresariales.
- Su aplicación abarca sectores como la banca, salud, gobierno, educación y defensa donde la seguridad y una correcta asignación de accesos son críticas.

MONITOREO DE INTEGRIDAD DEL SISTEMA

- Asegura que los archivos del sistema y los recursos críticos no hayan sido alterados.
- El objetivo principal es detectar cambios inesperados o NO AUTORIZADOS.
- Herramientas como AIDE, Tripwire y Samhain crean una línea base de la integridad de los archivos del sistema y luego se hacen comprobaciones periódicas para detectar cualquier desviación de esta línea base.
- Proporciona alertas y reportes que permiten tomar medidas.

ANALISIS DE VULNERABILIDADES

- Identificar, clasificar y priorizar vulnerabilidades en el sistema
- Se utilizan herramientas como OpenVAS, Nessus y Lynis para escanear el sistema en busca de vulnerabilidades conocidas.
- Genera informes y a menudo se ofrecen recomendaciones para mitigar los problemas identificados.

NMAP

- Acrónimo de Network MAPper
- Principal función es detectar dispositivos conectados y los servicios que están ejecutando.
- Su eficacia radica en su capacidad para enviar y analizar distintos tipos de paquetes.
- Puede detectar no solo dispositivos conectados, además puede detectar el tipo del SO, versiones del servicios específicos en los puertos abiertos, presencia de sistemas de filtrado como firewall.
- Es altamente configurable.

COMANDO	LO QUE HACE
nmap 192.168.1.105	Escaneo de los 1000 puertos TCP más comunes en un dispositivo ESPECIFICO.
nmap -sn 192.168.0.0/24	Descubre hosts activos (sin puertos) por medio de PING. Su objetivo solo es ver que direcciones IP estan activas.
nmap -p 22,80,443 192.168.0.0/24	Escanea los puertos 22 (SSH), 80 (HTTP), 443 (HTTPS) de cada IP. • No necesitas -sn, ya que nmap detecta automáticamente si el host está activo antes de escanear.
nmap -p- 192.168.0.0/24	MUCHO MAS LENTO!!! Pero escanea todos los puertos
nmap -p- 192.168.0.0/24 --stats-every 10s	mostrará el progreso cada 10 segundos. Se puede configurar para cambiar esos segundos, ideal para escaneos largos.
nmap -F 192.168.0.0/24	Escaneo rápido de todos los puertos comunes
nmap -Sv 192.168.1.105	determina que versión especifica de software está corriendo en los puertos abiertos.

PRUEBAS DE PENETRACION

- Estas pruebas se apoyan en herramientas como:
  - Nmap.
    Escaneo de puertos e identificación de servicios.
  - Metasploit
    Framework que permite el desarrollo y ejecución de exploits contra una maquina remota.
  - Wireshark
    Analizador de protocolos para entender el tráfico de red.
  - Aircrack-ng
    Pruebas de penetración en redes wifi.
  - John the Ripper
    Utilizado para identificar contraseñas débiles.

GESTION DE INCIDENTES DE SEGURIDAD

- Implica una serie de pasos y herramientas específicas para identificar, analizar y remediar vulnerabilidades y ataques.
- Los pasos de la gestión de incidentes son:
  - PREPARACION
    - Incluye:
      - Políticas y procedimientos.
        Como se gestionan los incidentes.
      - Herramientas de monitoreo y detección.
        IDS, IPS. Firewall, SIEM, etc.
      - Copias de seguridad.
        Regulares y actualizadas.
      - Formación del equipo.
        Equipo bien entrenado en la respuesta.
  - IDENTIFICACIÓN
    - Es el primer paso para resolverlo.
    - Herramientas y procedimientos:
      - Monitoreo de sistemas y redes.
        - Audit
        - Syslog
      - Analizar logs de sistemas y aplicaciones.
  - CONTENCION
    - Una vez identificado *evitar que se propague* o cause más daño.
    - Medidas:
      - Aislamiento de sistemas afectados.
      - Aplicación de parches o correcciones.

- ERRADICACION
  - Después de ser contenido, erradicar la causa raíz para evitar que se repita.
  - Medidas:
    - Eliminación de malware.
    - Reparación de vulnerabilidades.
- RECUPERACION
  - Volver a poner en servicio los sistemas afectados.
  - Incluye:
    - Restauración de datos desde copias de seguridad.
    - Monitoreo post-recuperación.
- APRENDIZAJE Y PREVENCION
  - Revisión y análisis del incidente.
  - Actualización del plan de respuesta a incidentes.
  - Formación continua.

COMANDOS Y HERRAMIENTAS DE SEGURIDAD PARA HARDENING EN SO OPEN SOURCE – SEGURIDAD DE RED

HERRAMIENTAS IPTABLES Y FIREWALLD

a. iptables

- Opera en la capa de red del modelo OSI.
- Herramienta de usuario para configurar el firewall del kernel de Linux.
- Utiliza tablas que tienen cadenas y las cadenas contienen un conjunto de reglas. Reglas que se utilizan para determinar cómo se manejan los paquetes de datos.
- Es parte de la suite de herramientas Netfilter.
- Entre las varias tablas que utiliza iptables están:
  - Filter

    La tabla predeterminada, se usa para permitir o bloquear paquetes.
  - NAT

    Traducciones de direcciones de red.
  - Mangle

    Para modificar paquetes, por ejemplo cambiar bits de ToS/DSCP.
  - Cadenas

    Las principales son:
    - *INPUT*
    - *OUTPUT*
    - *FORWARD*
  - Reglas

    Cada regla dentro de una cadena especifica como tratar los paquetes que coinciden con sus criterios, como son *aceptar*, *rechazar o descartar* los paquetes.

b. firewalld

- Opera en las capas de red y transporte del modelo OSI.
- Actúa como una interfaz para iptables.
- Simplifica el uso de iptables, permitiendo configurar el firewall de forma más fácil y flexible usando zonas (NIVELES DE CONFIANZA) y servicios (como HTTP, SSH) sin tener que escribir reglas manuales complicadas.
- Está diseñado para manejar cambios en la configuración del firewall sin tener que reiniciar el servicio, lo que es muy útil en entornos dinámicos y con cambios frecuentes.

c. Principios teóricos clave

- Modelo en capas.
  Tanto iptables como firewalld operan considerando el modelo de capas de red.
  Permite filtrar el tráfico según IP, puertos, protocolos, etc. respetando el modelo OSI.
- Estado de conexión.
  Las reglas pueden cambiar según si la conexión es nueva, existente, relacionada o invalida.
- Abstracción.
  firewalld hace más fácil usar iptables al permitir configurar reglas usando zonas y servicios sin tanto detalle técnico.

Todo esto ayuda a configurar un firewall más seguro y con mejor control.

REGISTRO DE AUDITORIA DEL SISTEMA

- Parte crucial de la seguridad y administración del sistema debido a que permite rastrear y registrar actividades del sistema.
- Lo anterior es útil no solo como herramienta de detección y diagnostico sino que también permite cumplir con requisitos de cumplimiento de seguridad y auditorias.

GESTION DE PAQUETES Y ACTUALIZACIONES EN ENTORNOS OPEN SOURCE.

- Fundamental mantener el software actualizado.
- Conceptos claves asociados con la gestión de paquetes y actualizaciones:
  - *Gestor de paquetes*
    - Colección de herramientas de software que automatiza el proceso de instalación, actualización, etc.
    - Los más comunes en entornos Linux incluyen:
      - apt (Debian y Ubuntu)
      - yum y dnf (Red Hat)
      - pacman (Arch Linux)
  - *Repositorios*
    - Almacenes de paquetes.
    - Pueden ser oficiales o de terceros.
    - Aseguran que los paquetes han sido previamente compilados y empaquetados para una fácil instalación.
  - *Dependencias*
    - Paquetes que deben estar presentes para que otros funcionen.
    - Los gestores de paquete manejan automáticamente las dependencias.
  - *Actualizaciones e instalación*
    - Proceso que incluye la actualización de las librerías del sistema, aplicaciones instaladas y el propio kernel de Linux si es necesario.
  - *Seguridad*
    - Principal razón para mantener software actualizado.

- *Rollback*
  - Algunos sistemas de gestión de paquetes permiten revertir las actualizaciones en caso de problemas de estabilidad.
- *Automatización*
  - Sistemas se pueden configurar para buscar, descargar e instalar de forma automática las actualizaciones.
- *Distribuciones de software LIBRE v/s PROPIETARIO*
  - En sistemas de código abierto, la mayoría del software es libre, pero también puede haber software propietario.
  - El sistema de paquetes permite elegir qué instalar, combinando libre y propietario según las necesidades.

SEGURIDAD EN LA GESTION REMOTA DE SISTEMAS Y LOS COMANDOS ASOCIADOS A LA CONFIGURACION DE SSH

- Administración de sistemas de forma remota y segura por medio de comunicación CIFRADA.
- SSH es la actualización segura de *Telnet* y *rlogin*.
- Conceptos clave:
  - Cifrado
    - Se utiliza cifrado de clave pública.
    - Protege los datos que viajan entre cliente y servidor.
  - Autenticación
    - Uso de contraseña
    - Uso de clave publica (MAS SEGURA)
  - Integridad de datos
    - Asegura que los datos no son alterados durante la transmisión mediante uso de autenticación de mensajes.
- Configuración y comandos básicos.
  - *ssh usuario@servidor:* Conecta al servidor.
  - *ssh-keygen:* Crea claves pública/privada.
  - *ssh-copy-id:* Copia la clave pública al servidor.
  - *ssh-add:* Añade claves al agente.
  - *ssh-agent:* Programa auxiliar que guarda y gestiona claves privadas.
  - *sshd_config:* Archivo que define cómo se comporta el servidor SSH.
- Mejores prácticas de seguridad para ssh.
  - Desactivar la autenticación por contraseña
    - Debido a ataques de fuerza bruta se recomienda utilizar la autenticación basada en claves.

- Desactivar el puerto por defecto (22)
  - *ayuda* a evitar ataques automatizados.
- Limitar acceso ssh
  - Con allowusers o allowgroups restringir el acceso solo a usuarios específicos.
- Usar claves fuertes
  - Para clave publica se debe generar claves de 2048 bits o mas
- Implementar Fail2ban o similares
  - Herramientas que ayudan a proteger de ataques de fuerza bruta.

SEGURIDAD EN LAS CONEXIONES HACIA EL SISTEMA OPERATIVO A TRAVES DE UNA HERRAMIENTA PARA CREAR VPN´s

- Configuración de VPN utilizando una herramienta de código abierto.
  - Se puede utilizar una herramienta de código abierto como:
    - OpenVPN
    - WireGuard
  - El proceso de configuración de una vpn:
    - *Selección de la herramienta*
      - Evaluar en función de la seguridad que ofrece, su facilidad de configuración, rendimiento y compatibilidad con los SO de los clientes y servidores.
    - *Instalación y configuración del servidor*
      - Instalar el software de la VPN en el servidor. Esto puede requerir dependencias adicionales y ajustes en la configuración del sistema operativo para habilitar el enrutamiento y el reenvío de paquetes.
    - *Configuración de Parámetros de Red*

      Establecer la configuración de red necesaria para la VPN, incluidos los rangos de IP para los clientes, las reglas de enrutamiento para dirigir el tráfico a través de la VPN y las configuraciones de NAT (Network Address Translation) si son necesarias.
    - *Generación de Credenciales*

      Crear un conjunto de claves y certificados para el servidor y para cada cliente. Esto generalmente incluye una Autoridad Certificadora

(CA), un certificado de servidor y claves privadas y públicas para cada cliente.

- *Configuración del Cliente*

  Instalar y configurar el software cliente en los dispositivos que se conectarán a la VPN, lo cual incluye importar las credenciales y configurar los parámetros específicos de conexión como la dirección del servidor, el puerto y el tipo de cifrado.

- Autenticación y cifrado de la conexión VPN
  - Autenticación
    - Se puede realizar mediante:
      - Certificados digitales.
      - Claves pre-compartidas.
      - Usuario y contraseña.
  - Cifrado
    - Los algoritmos de cifrado comunes incluyen AES (Advanced Encrytion Standard), por ejemplo AES 256.

- Medidas de seguridad adicionales en la conexión VPN
  - Además de autenticación y cifrado se puede implementar:
    - Firewall y reglas de acceso
      - Configurar para restringir el acceso a servicios y puertos específicos.
    - Autenticación de 2 factores (2FA)

        - Los usuarios deben proporcionar 2 formas de identificación antes de acceder a la VPN.
    - Seguridad a nivel de aplicación
        - Aplicar políticas de seguridad en las aplicaciones que se acceden a través de la VPN, como la autenticación en aplicaciones web.
    - Monitoreo y auditoría
        - Implementar herramientas de monitoreo.

- Trafico de la conexión VPN
    - Enrutamiento del trafico
        - Configurar para que todo el tráfico pase a través de la VPN o establecer rutas específicas como “SPLIT TUNNELING”.
    - Compresión y Optimización
        - Algunas herramientas de VPN permiten la compresión de datos para mejorar la velocidad de transmisión, aunque esto puede ser contraproducente si los datos ya están cifrados, ya que la compresión de datos cifrados puede no ser eficiente.
    - Calidad de Servicio (QoS)
        - Implementar políticas de QoS para priorizar ciertos tipos de tráfico y asegurar el ancho de banda necesario para aplicaciones críticas, mejorando la experiencia del usuario.


utsdirbih
cyberseccurity
Made by OZonE
benapablaza
benapablaza
benapablaza