Linuxprivchecker.py

Un script de verificación de escalamiento de privilegios de Linux

Este script está destinado a ser ejecutado localmente en una caja Linux para enumerar información básica del sistema y Búsqueda de vectores de escalada de privilegios comunes como archivos de escritura mundial, configuraciones erróneas, texto claro Contraseñas y exploits aplicables.

Linuxprivchecker está diseñado para identificar áreas potenciales para investigar más a fondo, no proporcionar acción directa o explotación. Esto es para ayudar a los usuarios a aprender más cómo funcionan estas escaladas de privilegios y mantenerlo en línea con las reglas, Para la explotación autodirigida, establecido para el OSCP, HTB y otros CTF/exámenes.

Ejecución en el sistema Legacy Python 2.6/2.7

Para ejecutar en los sistemas python >2.6 heredados, solo tiene que obtener el script de Python todo en uno y ejecutarlo.

```
wget
https://raw.githubusercontent.com/sleventyeleven/linuxprivchecker/master/linu
xprivchecker.py

python linuxprivchecker.py -w -o linuxprivchecker.log
```

Ejecutar en Current Python 3. Sistema X (Beta)

En este momento Linuxprivchecker para python 3. X debe considerarse una versión beta estable. Los problemas pueden ocurrir con el script y ciertamente puede perder posibles vulnerabilidades

Paso 1: Instalar pipx

Si aún no lo tienes, instálalo:

```
sudo apt update

sudo apt install pipx -y

pipx ensurepath
```

El comando pipx ensurepath se utiliza para **asegurar que el directorio donde pipx instala los ejecutables (scripts) esté incluido en la variable de entorno del sistema PATH.**

Esto es crucial para que puedas ejecutar los programas instalados con pipx directamente desde cualquier terminal o Símbolo del sistema, sin tener que escribir la ruta completa de la instalación.

Después de ejecutar pipx ensurepath, a menudo tendrás que **cerrar y volver a abrir la terminal** para que los cambios en la variable PATH surtan efecto.

Paso 2: Instalar LinuxPrivChecker con pipx

Una vez que pipx esté configurado, puedes instalar la aplicación directamente. pipx se encargará de crear y gestionar el entorno virtual por ti.

```
pipx install linuxprivchecker
```

Paso 3: Ejecutar la Herramienta

Una vez instalado, el comando linuxprivchecker debería estar disponible directamente en tu terminal.

Si no hicimos el paso anterior:

```
source ~/.bashrc
```

El comando source ~/.bashrc se utiliza para **cargar y ejecutar** el contenido del archivo de configuración de Bash (.bashrc) en la sesión de *shell* actual.

Cuando inicias un terminal, el archivo .bashrc se ejecuta automáticamente. Sin embargo, si has realizado **cambios** en ese archivo (como añadir nuevos *alias*, funciones o modificar la

variable $PATH) **después** de que la terminal se inició, estos cambios no se aplicarán inmediatamente.

Para ejecutarlo

linuxprivchecker

sudo linuxprivchecker

La variable de entorno PATH que usa el comando sudo (cuando ejecuta algo como *root*) **no incluye** la ruta donde pipx instaló el ejecutable linuxprivchecker.

¿Error?

1. **pipx instala el script:** pipx instaló linuxprivchecker en un directorio específico de tu usuario (algo como /home/feval/.local/bin o similar).
2. **Tu PATH de usuario:** Ejecutaste source ~/.bashrc, lo que probablemente asegura que **tu usuario (feval)** tenga la ruta de pipx en su PATH. Por eso, si hubieras escrito linuxprivchecker (sin sudo), habría funcionado.
3. **El PATH de sudo:** Cuando usas sudo, el sistema usa una variable PATH más restringida y segura (la de *root*), que **no incluye** rutas personales de usuario como el directorio de pipx. Por lo tanto, sudo no sabe dónde encontrar el programa llamado linuxprivchecker.

Solución: Usar la Ruta Completa

La forma más segura y rápida es decirle a sudo **exactamente dónde** está el programa.

El ejecutable instalado por pipx suele estar en ~/.local/bin/linuxprivchecker. Puedes usar la variable $HOME para esto:

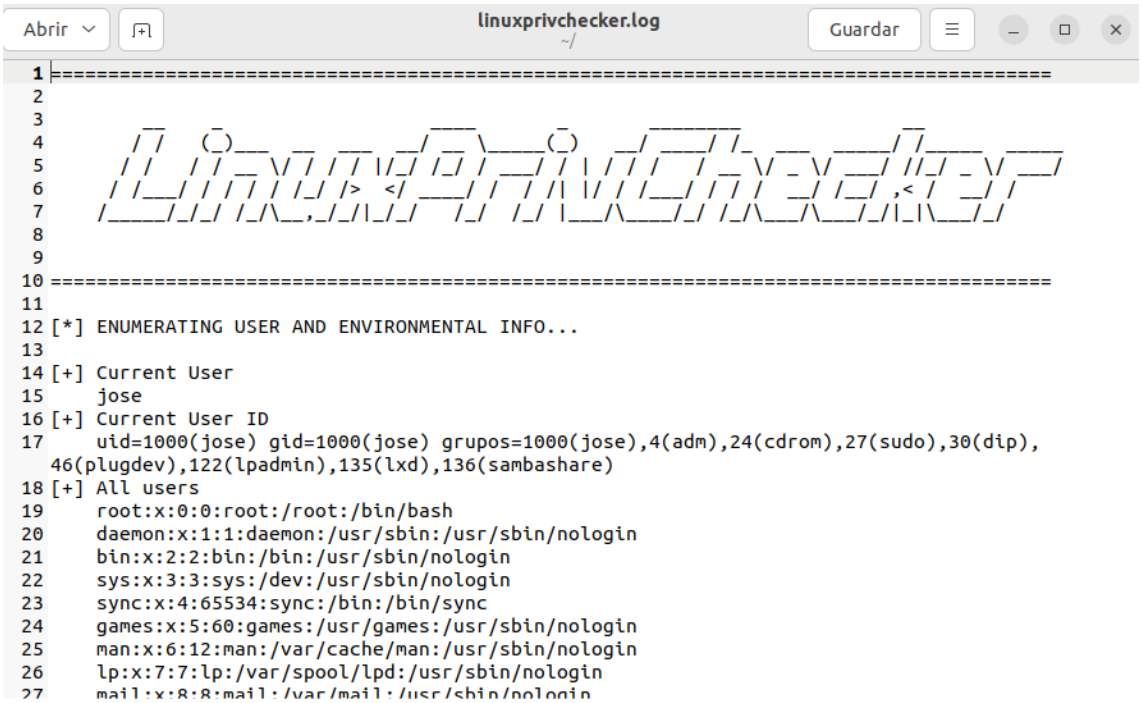
sudo $HOME/.local/bin/linuxprivchecker

2. Ejecutar con Permisos de tu Usuario (Si es posible)

Si el objetivo de la herramienta es solo inspeccionar, a veces puede ejecutarse primero sin sudo para ver si arroja alguna información útil, aunque las herramientas de auditoría suelen necesitar *root*:

linuxprivchecker

sudo $HOME/.local/bin/linuxprivchecker



Opciones de comando y argumentos

Esta herramienta, aunque es un *script* simple de Python, es fundamental para la fase de **enumeración local** en un sistema Linux. Su objetivo principal es automatizar la búsqueda de los errores de configuración más comunes que permiten la **escalada de privilegios** (pasar de un usuario de bajo nivel a *root*).

1. Sintaxis Básica y Opciones

El uso más común (y recomendado) es ejecutarlo con sudo para que pueda acceder a archivos protegidos y obtener una vista completa del sistema:

Comando	Descripción
sudo linuxprivchecker	**Uso estándar.** Ejecuta el escaneo completo en modo interactivo y genera la salida directamente en la terminal.
linuxprivchecker	Ejecuta el escaneo con los permisos del usuario actual. Solo devolverá información accesible por ese usuario, lo cual es útil para simular un atacante sin privilegios.

2. Opciones Útiles (Parámetros)

Aunque LinuxPrivChecker es bastante simple y no tiene una gran cantidad de argumentos, puedes usar estas opciones básicas de Python para modificar la ejecución:

Opción	Comando de Ejecución	Función
Ayuda	linuxprivchecker -h	Muestra un breve resumen de las opciones disponibles y la sintaxis.
Redirección a Archivo	*linuxprivchecker > /tmp/reporte.txt*	**Crucial:** Redirige toda la salida (el informe) a un archivo de texto para su análisis posterior. Esto es fundamental para el análisis forense y la documentación.
Ejecución Lenta	sudo python3 linuxprivchecker.py	Si la versión pipx da problemas, puedes ejecutar el archivo python directamente si lo clonaste, lo cual es útil para depuración.

cat /tmp/reporte.txt

sudo cat /tmp/reporte.txt

La forma más segura y rápida de ejecutar una herramienta instalada por un usuario con sudo es decirle a sudo la **ruta exacta** donde se encuentra el ejecutable:

sudo /home/feval/.local/bin/linuxprivchecker > /tmp/reporte.txt

3. Flujo de Trabajo y Análisis del Informe

El informe generado por LinuxPrivChecker se divide en secciones que analizan diferentes vectores de ataque. El objetivo no es solo ver la salida, sino saber qué buscar.

A. Información del Sistema y del Kernel

- **Busca:** Versión del Kernel (uname -a), Distribución, y Arquitectura.
- **Significado:** Si detecta un Kernel antiguo o una versión que ha sido marcada como vulnerable, indicará automáticamente el **CVE** (Common Vulnerabilities and Exposures) correspondiente, lo que sugiere un ataque conocido de explotación del Kernel.

B. Usuarios, Grupos y Privilegios Sudo

- **Busca:** Usuarios con UID 0 (root), miembros del grupo sudo o wheel, y configuraciones débiles en /etc/sudoers.
- **Significado:** Revela qué usuarios tienen más poder y qué comandos pueden ejecutar sin contraseña. La herramienta destaca especialmente las líneas en /etc/sudoers que permiten a un usuario ejecutar un comando peligroso como find, vim, o nmap con sudo, lo cual se puede explotar fácilmente para obtener un *shell* de *root*.

C. Servicios y Procesos

- **Busca:** Procesos en ejecución con privilegios de root (ps aux), servicios inusuales, y archivos de configuración sensibles.
- **Significado:** Un proceso ejecutándose como *root* que tiene una versión vulnerable o una configuración incorrecta es un objetivo directo para la explotación.

D. Archivos y Permisos (El Punto Débil Común)

Esta es la sección más importante para la escalada de privilegios en Linux.

Vector de Ataque	**Archivos a Revisar**	**Potencial de Escalada**
Archivos SUID/SGID	Archivos con los *bits* SUID/SGID establecidos (ej.	Un binario con SUID permite a cualquier usuario ejecutarlo con los

Vector de Ataque	Archivos a Revisar	Potencial de Escalada
	/usr/bin/find, /usr/bin/nmap).	permisos del dueño (si el dueño es root, ¡el usuario se convierte en *root*!). La herramienta lista binarios SUID peligrosos.
Archivos de Configuración Escribibles	*Scripts* de servicio (/etc/init.d/) o archivos de *cron* que el usuario actual puede modificar.	Si puedes modificar un *script* que se ejecuta como *root* (ej. al reiniciar un servicio), puedes añadir código malicioso para obtener un *shell* de *root*.
Contraseñas Almacenadas	Archivos de configuración o *scripts* que contienen contraseñas codificadas o en texto plano.	Permite saltarse la autenticación o acceder a otros servicios con los credenciales del administrador.

E. Entorno y Variables

- **Busca:** Variables de entorno sensibles (como LD_PRELOAD) y *paths* débiles.
- **Significado:** Algunas vulnerabilidades de escalada de privilegios se basan en manipular las librerías que carga un programa o las rutas de búsqueda para inyectar código malicioso.

4. Estrategia de Hardening (Mejoras de Seguridad)

Una vez que tengas el informe, tu misión será corregir cada punto de riesgo.

1. **Vulnerabilidades del Kernel:** Si se detecta un CVE, **actualiza el sistema operativo inmediatamente** (sudo apt update && sudo apt upgrade).
2. **Archivos SUID/SGID: Elimina el *bit* SUID** de binarios que no lo necesiten de forma crítica (ej. sudo chmod u-s /usr/bin/find).

3. **Archivos Escribibles:** Restringe los permisos de escritura a archivos sensibles (/etc/passwd, *scripts* de *cron*, archivos de configuración de servicios) para que solo *root* pueda modificarlos (sudo chmod 600 <archivo>).
4. **Limpieza:** Elimina cuentas de usuario innecesarias y cualquier *script* de *cron* que se ejecute como *root* pero que no sea esencial.
5. **Políticas Sudoers:** Asegúrate de que las reglas de sudoers sean lo más estrictas posible, evitando la directiva NOPASSWD y permitiendo solo comandos específicos.

Resultado

Resumen y Puntos Clave para la Escalada de Privilegios

1. Información de Usuario y Privilegios (Máxima Prioridad)

- **Grupos del Usuario Actual (feval):**
    - El usuario feval pertenece al grupo **sudo**.
    - **Significado:** Este es el vector de escalada más obvio y directo. Si el usuario feval conoce su contraseña, puede ejecutar cualquier comando como root usando sudo, lo que esencialmente significa que **ya tienes control de root** a través de este mecanismo.
- **Archivos de Historial del Usuario (.bash_history):**
    - El historial de comandos revela información sensible o comandos clave. En el archivo se ven rastros de la instalación y configuración de un entorno **LAMP (Linux, Apache, MySQL, PHP)**.
    - **Puntos de Interés:**
        - Comandos relacionados con **MySQL** (sudo mysql_secure_installation, sudo mysql). Si el historial hubiera incluido una contraseña, sería una exposición directa.
        - Configuración de **WordPress** (sudo nano /etc/apache2/sites-available/wordpress.conf, sudo chown -R www-data:www-data /var/www/wordpress, sudo nano wp-config.php). El archivo wp-config.php a menudo contiene credenciales de bases de datos codificadas.

2. **Información del Sistema y Aplicaciones (Vulnerabilidades)**

- **Versión del Kernel ([*] GETTING BASIC SYSTEM INFO...):**
    - El sistema está ejecutando **Ubuntu 24.04.3 LTS** con el kernel **Linux version 6.14.0-33-generic**.
    - **Significado:** Esta información es vital. Si no puedes usar sudo (o si ya lo usaste, pero buscas un método alternativo), buscarías vulnerabilidades (exploits) conocidas para esta versión específica del Kernel.
- **Procesos Ejecutándose como root:**
    - El reporte lista procesos de alto privilegio.
    - **Puntos de Interés:**
        - **apache2** está corriendo como root[6]. Si el servidor web está mal configurado y permite la carga de archivos, podría explotarse. El historial confirma que hay una instalación de **WordPress**, lo que aumenta la superficie de ataque.
        - Otros procesos como mysql están presentes. Si pudieras interactuar con el servicio MySQL y reescribir archivos binarios del sistema, podrías escalar privilegios (por ejemplo, mediante una UDF maliciosa).

3. **Configuraciones y Archivos del Sistema**

- **Archivos de Configuración (*.rc Style Files):**
    - El archivo **.bashrc** muestra que se agregó la ruta del usuario para pipx (/home/feval/.local/bin) a la variable $PATH[9].
    - **Significado:** Si el sistema ejecuta un script con root que invoca un comando simple (sin una ruta absoluta) que el atacante pudiera recrear y colocar en /home/feval/.local/bin, se podría ejecutar código como root.
- **Puertos y Servicios de Red ([*] GETTING NETWORKING INFO...):**
    - El sistema tiene abiertos los siguientes puertos para escucha:
        - **Puerto 80 (HTTP)** en IPv6 (Web Server).

        - **Puerto 3306** (MySQL).
    - **Significado:** Los servicios expuestos son posibles puntos de entrada. En particular, la instalación de MySQL y Apache/WordPress, junto con la información del historial, dirige tu atención a estas aplicaciones para buscar credenciales o inyecciones.

4. Búsqueda de Archivos (Pasos Manuales)

- **Comandos en el Historial:** El historial revela que el usuario ha buscado directorios con permisos de escritura universales:
-

```
find / -type d -perm 777
```

```
sudo find / -type d -perm 777
```

    - **Significado:** Esta es una técnica estándar de escalada. El reporte de linuxprivchecker continúa buscando esto, pero el historial confirma que el usuario es consciente de esta técnica. Si el reporte encuentra directorios o archivos de configuración críticos que el usuario feval pueda escribir, esa es una oportunidad de escalada.