Reporte Ejecutivo de Ciberseguridad: Análisis de Vulnerabilidad y Explotación en el Sistema "LinkVortex"

---

1. Portada

- **Título del Reporte**: Análisis de Vulnerabilidad y Explotación en el Sistema "LinkVortex"
- **Fecha**: 12 de Diciembre de 2024
- **Nombre de la Organización**: (N/A) - Proyecto Educativo
- **Autor(es)**: tellmefred

---

2. Resumen Ejecutivo

El presente informe documenta el proceso de análisis y explotación de la máquina "LinkVortex" de Hack The Box. Esta máquina de dificultad fácil representa un entorno que simula vulnerabilidades comunes en sistemas web y servicios internos. A través de técnicas de enumeración de subdominios, análisis de repositorios `.git` y explotación de servicios expuestos, se logró obtener acceso a credenciales sensibles, escalando privilegios hasta obtener acceso root.

Este reporte detalla cada fase del proceso, proporcionando un análisis técnico, recomendaciones de mitigación y buenas prácticas de seguridad para prevenir ataques similares.

---

3. Introducción

- **Contexto**: "LinkVortex" es una máquina diseñada para practicar la explotación de servicios web y elevación de privilegios en entornos controlados.
- **Propósito**: Desarrollar y reforzar habilidades técnicas en reconocimiento, explotación de servicios y escalada de privilegios.
- **Alcance**: El análisis abarca desde el reconocimiento inicial hasta la obtención de acceso root, incluyendo la explotación de servicios web y configuración insegura de permisos en scripts de sudo.
- **Metodología**: Se emplearon herramientas de escaneo (`nmap`), búsqueda de directorios (`gobuster`) y técnicas de análisis de repositorios Git expuestos para acceder a información confidencial.

---

4. Estado Actual de la Ciberseguridad

- **Resumen de la Postura de Ciberseguridad**:
  La máquina "LinkVortex" presentó múltiples vulnerabilidades explotables:
  - Exposición de un repositorio Git en producción.
  - Almacenamiento inseguro de credenciales en archivos de configuración.
  - Permisos inseguros en scripts que permitieron escalada de privilegios.
- **Sistemas y Datos Críticos**:
  - Servicio web en el puerto 80.
  - Servicio SSH en el puerto 22.
  - Repositorio Git expuesto con credenciales en archivos de configuración.
  - Acceso privilegiado a través de scripts ejecutables por sudo.

---

5. Evaluación de Vulnerabilidades y Explotación

Reconocimiento

1. **Escaneo de Red**:
   - Se realizó un escaneo `nmap` que reveló los puertos 22 (SSH) y 80 (HTTP) abiertos.
   - Se agregó el dominio de la máquina al archivo `/etc/hosts` para facilitar el acceso web.
2. **Enumeración de Subdominios**:
   - Se identificó un subdominio `.dev` que fue agregado al `/etc/hosts` para su análisis.
   - Se utilizó `gobuster` para realizar una búsqueda de directorios, revelando un directorio `.git`.
3. **Análisis de Repositorio Git**:
   - Se descargó el contenido del repositorio expuesto usando `git dumper` para analizar el código fuente en busca de credenciales.
   - Se identificó un archivo con credenciales que permitieron el acceso al dashboard del sistema.

Explotación

1. **Explotación de Archivos Sensibles**:
   - Se identificó una vulnerabilidad que permitió visualizar archivos internos del sistema.
   - Mediante el análisis del Dockerfile de Ghost, se localizó una ruta clave que contenía configuraciones con credenciales sensibles.
2. **Acceso Inicial**:
   - Las credenciales obtenidas se utilizaron para acceder al sistema y capturar la bandera `user.txt`.

Escalada de Privilegios

1. **Permisos de Sudo**:
   - Se ejecutó `sudo -l` para analizar permisos del usuario. Se detectó que el usuario tenía permisos para ejecutar un script como root.
2. **Análisis del Script**:
   - Tras analizar el script, se utilizó para capturar el archivo `id_rsa` del usuario root.
   - Se accedió al sistema como root y se capturó la bandera `root.txt`.

---

6. Recomendaciones

1. **Protección de Repositorios Git**:
   - Asegurar que los repositorios Git no estén expuestos en entornos de producción.
   - Implementar políticas de acceso restringido y eliminar archivos sensibles antes de la implementación.
2. **Gestión Segura de Credenciales**:
   - Evitar almacenar credenciales en archivos de configuración o código fuente.
   - Utilizar gestores de secretos para proteger contraseñas y claves API.
3. **Configuración Segura de Permisos**:
   - Revisar los scripts ejecutables con permisos sudo y aplicar el principio de mínimo privilegio.
   - Limitar el uso de `sudo` únicamente a las operaciones necesarias y auditar periódicamente los permisos.
4. **Segmentación de Servicios**:
   - Asegurar que los servicios internos (por ejemplo, Dockerfiles) no expongan rutas o configuraciones sensibles.
   - Limitar el acceso a servicios web solo desde direcciones IP autorizadas.

---

7. Conclusión

El análisis de la máquina "LinkVortex" reveló vulnerabilidades críticas que permitieron comprometer completamente el sistema. Este escenario resalta la importancia de implementar buenas prácticas de desarrollo seguro, control de acceso y gestión adecuada de credenciales. La aplicación de las recomendaciones proporcionadas permitirá mitigar riesgos similares en futuros entornos.

---

8. Anexos

- Escaneos de red (`nmap`) y resultados de enumeración de directorios.
- Capturas de pantalla del proceso de explotación y escalada de privilegios.
- Detalles técnicos sobre el análisis del repositorio Git expuesto.