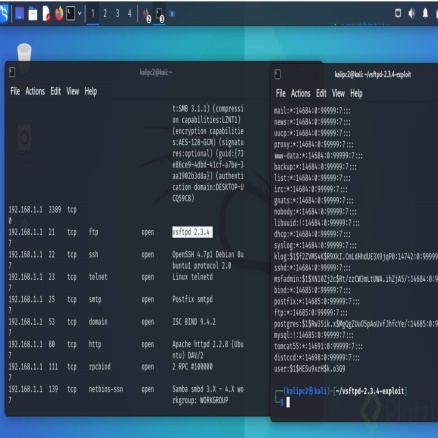
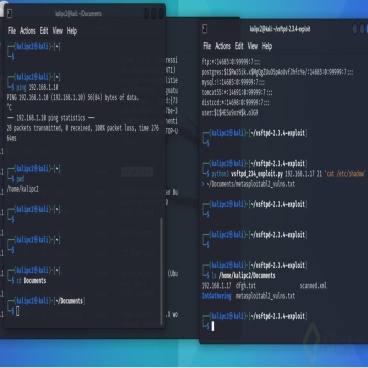
<https://github.com/ahervias77/vsftpd-2.3.4-exploit>

, \nListo, Actividad/Tarea Completa el hardcodedBackdoor de my Metasploitable2 VM.\n



\nLuego Yo use el Command :\t python3 vsftpd\_234\_exploit.py 192.168.20.46 21 "cat /etc/shadow" > **-/Documents/Metasploitable2\_vulns.txt para guardar un .txt file con los resultados del fichero /etc/shadow de la maquina metasploitable2**



Una "backdoor" (puerta trasera) hardcoded es un acceso no autorizado y predefinido dentro de un software o sistema, generalmente insertado por el desarrollador o atacante, que permite el acceso al sistema sin pasar por los mecanismos normales de seguridad. Este tipo de backdoor suele incluir credenciales o comandos que están codificados directamente en el software, lo que permite al atacante o al desarrollador acceder al sistema en cualquier momento.

**Características de una Hardcoded Backdoor**

1. **Credenciales Embebidas**:
   * Uso de nombres de usuario y contraseñas predefinidos que no pueden ser cambiados fácilmente por el usuario final.
2. **Acceso Oculto**:
   * Proporciona un método oculto para acceder al sistema, que no es evidente para los usuarios o administradores legítimos.
3. **Dificultad de Detección**:
   * A menudo son difíciles de detectar mediante métodos normales de auditoría y pueden requerir una revisión del código fuente para su identificación.

**Ejemplos y Riesgos**

1. **Contraseñas Hardcoded**:
   * Un ejemplo clásico es cuando una aplicación o dispositivo tiene una contraseña de administrador predeterminada que no puede ser cambiada, o es difícil de cambiar. Esto es común en algunos dispositivos IoT y sistemas embebidos.
2. **Comandos Especiales**:
   * Comandos secretos que activan modos de administración o depuración en software que no están documentados para el usuario final.
3. **Puertas Traseras en Firmware**:
   * Los dispositivos de red, como routers o cámaras IP, pueden tener backdoors embebidas en su firmware que permiten a los atacantes tomar control del dispositivo.

**Ejemplo de Código**

Aquí hay un ejemplo simplificado de cómo podría verse una backdoor hardcoded en un código de software:

python

Copy code

def authenticate(user, password):

# Backdoor hardcoded

if user == "admin" and password == "1234":

return True

# Método normal de autenticación

return check\_credentials\_in\_database(user, password)

En este ejemplo, cualquier persona que conozca el nombre de usuario "admin" y la contraseña "1234" puede acceder al sistema sin pasar por el proceso de autenticación normal.

**Mitigación y Buenas Prácticas**

1. **Revisión de Código y Auditorías**:
   * Revisar el código fuente de manera regular para asegurarse de que no existan credenciales hardcoded o métodos de acceso no documentados.
2. **Actualizar Credenciales**:
   * Permitir y requerir que los usuarios cambien las credenciales predeterminadas en dispositivos y aplicaciones.
3. **Seguridad en el Ciclo de Desarrollo**:
   * Implementar prácticas de desarrollo seguro, como la programación defensiva y la revisión de pares, para prevenir la introducción de backdoors.
4. **Monitoreo y Detección**:
   * Implementar sistemas de monitoreo que puedan detectar accesos no autorizados y comportamientos anómalos en el sistema.

**Ejemplo Práctico en Metasploit**

En Metasploit, puedes buscar y utilizar exploits que aprovechen backdoors hardcoded en dispositivos y software vulnerables. Por ejemplo, si conoces un backdoor en un dispositivo específico, puedes buscar un módulo en Metasploit que lo explote:

plaintext

Copy code

msf > search backdoor

Una vez encontrado el exploit adecuado, puedes configurarlo y ejecutarlo de manera similar a cualquier otro exploit en Metasploit.

**Uso Responsable**

Es crucial recordar que el uso de backdoors, incluso con fines educativos, debe realizarse en entornos controlados y con permiso explícito. Explorar y explotar vulnerabilidades sin autorización es ilegal y antiético.