Informe de Análisis de Vulnerabilidad

Detección y Corrección de Vulnerabilidad en Servidor FTP

Proyecto Final de Ciberseguridad para 4Geeks Academy

Preparado por: Riccardo Barbieri

Fecha: 28/03/2025

Introducción

El presente informe describe los resultados de las pruebas de penetración realizadas sobre el servidor crítico comprometido en 4Geeks Academy como parte de la Fase 2 del proyecto final de seguridad informática. Esta fase tiene como objetivo escanear, detectar y explotar una vulnerabilidad diferente a la encontrada en la Fase 1, documentando todo el proceso y las medidas correctivas aplicadas.

En este caso, tras realizar un análisis exhaustivo del sistema, se identificó que el servicio FTP del servidor presentaba una configuración vulnerable que permitía el acceso anónimo con capacidad de escritura. Esta vulnerabilidad, distinta a la explotada en la fase anterior, representa un riesgo significativo para la seguridad de la infraestructura de 4Geeks Academy.

La metodología empleada para esta evaluación está basada en estándares internacionales de seguridad, incluyendo OWASP (Open Web Application Security Project) y PTES (Penetration Testing Execution Standard), adaptados específicamente para abordar vulnerabilidades relacionadas con servicios de red como FTP, SSH y HTTP.

Objetivos

Este informe corresponde específicamente a la Fase 2 del proyecto final, donde se requiere:

1. Escanear el sistema para detectar una vulnerabilidad adicional, diferente a la explotada previamente.
2. Explotar dicha vulnerabilidad de manera controlada para entender su impacto.
3. Escalar privilegios utilizando la vulnerabilidad detectada.
4. Corregir la vulnerabilidad y documentar el proceso completo.

El objetivo principal del análisis fue identificar vulnerabilidades de alto impacto dentro del servidor crítico de 4Geeks Academy, centrándose en el servicio FTP anónimo, cuya explotación podría conducir a:

1. Acceso no autorizado a datos confidenciales
2. Ejecución remota de código (RCE)
3. Escalación de privilegios dentro del sistema
4. Compromiso total del servidor

Específicamente, los objetivos detallados incluían:

- Evaluar la configuración y seguridad del servicio FTP
- Identificar si el acceso anónimo estaba habilitado y qué recursos eran accesibles
- Determinar si existían posibilidades de carga de archivos
- Verificar la ejecución de archivos cargados mediante el servicio
- Evaluar las posibilidades de post-explotación
- Analizar las vías potenciales para escalar privilegios
- Proponer soluciones efectivas para corregir la vulnerabilidad identificada

Convenciones utilizadas para valorar y categorizar cada hallazgo

En este informe se utiliza el sistema de puntuación CVSS v3 (Common Vulnerability Scoring System), diseñado para proporcionar un método estándar y abierto que permite evaluar el impacto de las vulnerabilidades identificadas en tecnologías de la información. Este sistema ayuda a cuantificar la gravedad de estas vulnerabilidades, utilizando una escala que va de 0 a 10.

Según la puntuación obtenida mediante la fórmula CVSS, se puede determinar el nivel de impacto:

- Para puntuaciones entre 0.0 y 3.9, la severidad se considera baja.
- Para puntuaciones entre 4.0 y 6.9, la severidad es media.
- Para puntuaciones entre 7.0 y 8.9, la severidad es alta.
- Para puntuaciones entre 9.0 y 10.0, la severidad es crítica.

El vector CVSS proporciona información detallada sobre cómo se compone la puntuación, incluyendo factores como:

- Vector de ataque (AV): Red, Adyacente, Local, Físico
- Complejidad del ataque (AC): Baja, Alta
- Privilegios requeridos (PR): Ninguno, Bajo, Alto
- Interacción del usuario (UI): Ninguna, Requerida
- Alcance (S): Sin cambio, Cambiado
- Confidencialidad (C): Ninguna, Baja, Alta

- Integridad (I): Ninguna, Baja, Alta

Sin embargo, el sistema CVSS no tiene en cuenta ciertas características comerciales. Por ejemplo, en industrias como la bancaria o la aérea, que están sujetas a estrictos requisitos regulatorios, el rango de riesgo puede ser mayor. Esta categorización de riesgos puede ser revisada junto con el cliente y ajustada según sus necesidades específicas.

Metodología de Prueba

En esta evaluación se han utilizado metodologías y frameworks reconocidos a nivel mundial, con especial atención al análisis y explotación de vulnerabilidades de red y servicios expuestos. La metodología aplicada integra los aspectos más relevantes de varios enfoques, combinándolos en un proceso integral para probar y evaluar la seguridad del servicio FTP y los sistemas asociados.

El proceso de prueba ha seguido estas fases:

1. Reconocimiento e Inteligencia

- **OSINT (Open Source Intelligence)**: Recopilación de información disponible públicamente sobre el objetivo
- **Recopilación de Información Pasiva**: Análisis de dominios, subdominios, y servicios expuestos sin interactuar directamente con el objetivo
- **Recopilación de Información Activa**: Escaneo de puertos, servicios y sistemas operativos

2. Análisis de Vulnerabilidades

- **Enumeración de servicios**: Identificación detallada de los servicios en ejecución y sus versiones
- **Identificación de vulnerabilidades conocidas**: Búsqueda de CVEs y exploits conocidos para las versiones de software detectadas
- **Análisis de configuración**: Evaluación de configuraciones inseguras o por defecto

3. Explotación

- **Aprovechamiento de fallos de configuración**: Acceso a servicios mal configurados (como FTP anónimo)
- **Explotación de vulnerabilidades**: Uso de técnicas específicas para aprovechar las vulnerabilidades encontradas
- **Upload y ejecución de payloads**: Carga y activación de código malicioso para obtener acceso

4. Post-Explotación

- **Escalación de privilegios**: Búsqueda de métodos para aumentar los privilegios en el sistema
- **Persistencia**: Evaluación de métodos para mantener el acceso al sistema
- **Movimiento lateral**: Análisis de posibles rutas para comprometer otros sistemas de la red

5. Documentación y Reporte

- **Registro detallado**: Documentación de todos los hallazgos con evidencias
- **Evaluación de impacto**: Análisis del impacto potencial de cada vulnerabilidad
- **Recomendaciones de mitigación**: Propuestas específicas para corregir cada vulnerabilidad

Alcance

Para realizar el análisis de seguridad sobre el servidor crítico de 4Geeks Academy como parte de la Fase 2 del proyecto final, el alcance se ha definido de la siguiente manera:

- Dirección IP del servidor: 10.0.2.13
- Servicios principales a evaluar:
    - FTP (puerto 21) - Foco principal de la investigación en esta fase
    - SSH (puerto 22)
    - HTTP (puerto 80)
- Tipos de prueba autorizados como parte del proyecto:
    - Escaneo de puertos utilizando herramientas como Nmap
    - Enumeración de servicios para identificar versiones y configuraciones
    - Pruebas de acceso FTP anónimo
    - Carga de archivos en servicio FTP (para demostrar el impacto de la vulnerabilidad)
    - Evaluación de permisos de archivos y directorios
    - Ejecución controlada de código (reverse shell)
    - Evaluación y explotación de vulnerabilidades para escalación de privilegios
- Tipos de prueba excluidos:
    - Ataques de denegación de servicio (DoS)
    - Fuerza bruta extensiva que pudiera comprometer la disponibilidad del sistema
    - Modificación o eliminación de datos críticos del sistema

- Cualquier acción que pudiera interferir con la operación normal del servidor más allá de lo necesario para demostrar la vulnerabilidad

Resumen Ejecutivo

Durante la ejecución de la Fase 2 del proyecto final de seguridad informática para 4Geeks Academy, se ha detectado y explotado una vulnerabilidad crítica diferente a la identificada en la fase anterior. Este informe documenta el proceso completo de identificación, explotación y corrección de la vulnerabilidad encontrada.

Se ha identificado una vulnerabilidad crítica, **Ejecución Remota de Código via FTP Anónimo (CWE-434)**, que permite ejecutar comandos en el servidor de manera remota mediante la carga de archivos ejecutables a través del servicio FTP con credenciales anónimas. Esta vulnerabilidad es particularmente peligrosa porque no requiere autenticación y permite a un atacante obtener acceso inicial al sistema comprometiendo gravemente la seguridad de la infraestructura de 4Geeks Academy.

Además, se ha detectado una vulnerabilidad de impacto crítico **Escalación de Privilegios en Sistema Debian (CWE-269)** que permite tomar control del usuario administrador (root) en el servidor mediante la explotación de una vulnerabilidad del kernel. Esta combinación de vulnerabilidades posibilita el compromiso total del sistema.

También se ha identificado una vulnerabilidad de impacto medio denominada **Exposición de Datos Sensibles via FTP Anónimo (CWE-200)** que permite a usuarios no autorizados acceder a información potencialmente sensible alojada en el servidor FTP.

El impacto potencial de estas vulnerabilidades es severo, ya que la combinación de acceso anónimo al FTP, capacidad de carga de archivos ejecutables y posibilidad de escalación de privilegios permite a un atacante obtener control completo del servidor con privilegios administrativos en poco tiempo y con un esfuerzo mínimo, poniendo en riesgo la confidencialidad, integridad y disponibilidad de los datos y servicios de la academia.

Conclusiones Generales

Como parte de la Fase 2 del proyecto final de seguridad informática para 4Geeks Academy, se ha completado con éxito la identificación, explotación y documentación de una vulnerabilidad diferente a la encontrada en la fase anterior. Durante este proceso, se identificaron un total de tres vulnerabilidades, de las cuales dos fueron clasificadas como críticas y una como de nivel medio. Estas vulnerabilidades representan riesgos significativos para la seguridad y la

integridad del servidor crítico de la academia, y requieren una acción inmediata para su mitigación.

La vulnerabilidad más crítica encontrada es la configuración inadecuada del servicio FTP que permite acceso anónimo con capacidad de escritura. Esta configuración, combinada con la posibilidad de ejecutar archivos cargados y posteriormente escalar privilegios, crea un vector de ataque completo que puede llevar al compromiso total del sistema. Este hallazgo es especialmente relevante ya que proporciona una vía de acceso adicional y distinta a la identificada en la fase anterior del proyecto.

Para mitigar estos riesgos y asegurar el servidor de 4Geeks Academy, la prioridad principal debe ser:

1. Deshabilitar el acceso FTP anónimo o, como mínimo, restringirlo a solo lectura
2. Actualizar el kernel del sistema para parchear las vulnerabilidades que permiten la escalación de privilegios
3. Revisar los datos accesibles a través del servicio FTP para asegurar que no haya información sensible expuesta
4. Implementar un mecanismo de monitorización de logs para detectar intentos de acceso no autorizados

Estas medidas, cuando se implementen, proporcionarán una mejora significativa en la postura de seguridad del servidor, reduciendo el riesgo de compromiso a través de los vectores de ataque identificados en esta fase del proyecto.

Detalles Técnicos

En el transcurso de la sección de detalles técnicos, se explicará en qué consiste cada vulnerabilidad identificada durante la Fase 2 del proyecto y se desarrollará su proceso de explotación y concatenación. Este análisis detallado cumple con el requisito de documentar completamente el proceso de detección y explotación de una vulnerabilidad diferente a la identificada en la fase anterior.

Las vulnerabilidades descritas a continuación fueron descubiertas como parte del escaneo y análisis de seguridad realizado al servidor crítico de 4Geeks Academy, siguiendo la metodología definida para esta fase del proyecto. Para cada vulnerabilidad, se incluyen recomendaciones específicas para su mitigación, las cuales deberían implementarse como parte del proceso de aseguramiento del servidor.

Descripción

El File Transfer Protocol (FTP) es un protocolo de red estándar utilizado para la transferencia de archivos entre un cliente y un servidor en una red. El acceso

anónimo FTP permite a cualquier usuario autenticarse al servidor usando credenciales genéricas (típicamente "anonymous" como nombre de usuario y una dirección de correo electrónico como contraseña).

Cuando un servidor FTP está configurado para permitir el acceso anónimo con privilegios de escritura, un atacante puede cargar archivos arbitrarios, incluidos programas ejecutables o scripts maliciosos. Si estos archivos se ejecutan posteriormente en el servidor, el atacante puede obtener una shell y, en consecuencia, una ejecución remota de código (Remote Code Execution - RCE).

En el caso específico evaluado, el servidor FTP permitía:

1. Acceso anónimo sin restricciones
2. Escritura en la carpeta "upload"
3. Posibilidad de ejecutar los archivos cargados

Esta configuración incorrecta del servicio FTP crea un vector de ataque directo que permite a un atacante obtener acceso inicial al sistema sin necesidad de credenciales válidas.

Remediación

Como parte de la corrección de vulnerabilidades requerida en la Fase 2 del proyecto, se recomienda implementar las siguientes medidas de seguridad específicas para el servidor de 4Geeks Academy:

- **Deshabilitar completamente el acceso FTP anónimo** modificando el archivo de configuración de vsftpd (ubicado en /etc/vsftpd.conf) y estableciendo anonymous_enable=NO. El acceso anónimo no es necesario en un servidor crítico que gestiona servicios e información sensible de la academia.
- Si por algún requisito específico de negocio fuera absolutamente necesario mantener el acceso anónimo, entonces **configurarlo en modo estricto de solo lectura** modificando los siguientes parámetros:

```
anonymous_enable=YES
anon_upload_enable=NO
anon_mkdir_write_enable=NO
```

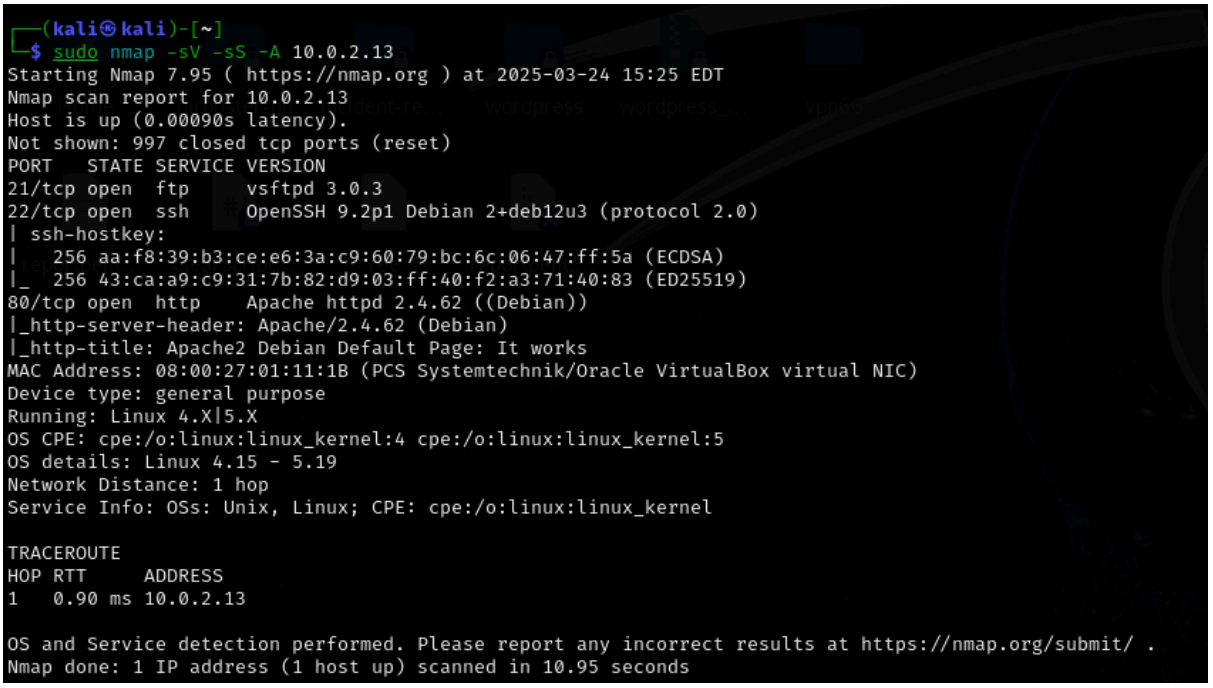
Explotación

Se ha identificado la vulnerabilidad **Ejecución Remota de Código via FTP Anónimo** que ha permitido ejecutar código de forma remota en la máquina víctima.

A continuación, se detalla paso a paso cómo se ha explotado esta vulnerabilidad:

Paso 1: Se ha realizado un escaneo completo de puertos en el servidor objetivo utilizando Nmap para identificar servicios disponibles:

nmap -sV -sS -A 10.0.2.13



El resultado del escaneo reveló que el servidor tiene el puerto 21 abierto con un servicio FTP activo, específicamente vsftpd 3.0.3:

Adicionalmente, se realizó un escaneo de vulnerabilidades utilizando scripts NSE (Nmap Scripting Engine) para identificar posibles vulnerabilidades en los servicios detectados:

nmap --script vuln 10.0.2.13

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV --script=vuln 10.0.2.13
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-28 14:10 EDT
Nmap scan report for 10.0.2.13
Host is up (0.00051s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| vulners:
|   vsftpd 3.0.3:
|       CVE-2021-30047  7.5     https://vulners.com/cve/CVE-2021-30047
|_      CVE-2021-3618   7.4     https://vulners.com/cve/CVE-2021-3618
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:9.2p1:
|       2C119FFA-ECE0-5E14-A4A4-354A2C38071A    10.0    https://vulners.com/githubexploit/2C119FFA-
ECE0-5E14-A4A4-354A2C38071A     *EXPLOIT*
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       CVE-2023-28531  9.8     https://vulners.com/cve/CVE-2023-28531
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/githubexploit/B8190CDB-
3EB9-5631-9828-8064A1575B23     *EXPLOIT*
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/githubexploit/8FC9C5AB-
3968-5F3C-825E-E8DB5379A623     *EXPLOIT*
|       8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/githubexploit/8AD01159-
548E-546E-AA87-2DE89F3927EC     *EXPLOIT*
|       5E6968B4-DBD6-57FA-BF6E-D9B2219DB27A    9.8     https://vulners.com/githubexploit/5E6968B4-
DBD6-57FA-BF6E-D9B2219DB27A     *EXPLOIT*
|       33D623F7-98E0-5F75-80FA-81AA666D1340    9.8     https://vulners.com/githubexploit/33D623F7-
98E0-5F75-80FA-81AA666D1340     *EXPLOIT*
|       0221525F-07F5-5790-912D-F4B9E2D1B587    9.8     https://vulners.com/githubexploit/0221525F-
07F5-5790-912D-F4B9E2D1B587     *EXPLOIT*
|       95499236-C9FE-56A6-9D7D-E943A24B633A    8.9     https://vulners.com/githubexploit/95499236-
C9FE-56A6-9D7D-E943A24B633A     *EXPLOIT*
|       PACKETSTORM:179290      8.1     https://vulners.com/packetstorm/PACKETSTORM:179290      *EX
PLOIT*
|       FB2E9ED1-43D7-585C-A197-0D6628B20134    8.1     https://vulners.com/githubexploit/FB2E9ED1-
43D7-585C-A197-0D6628B20134     *EXPLOIT*
|       FA3992CE-9C4C-5350-8134-177126E0BD3F    8.1     https://vulners.com/githubexploit/FA3992CE-
9C4C-5350-8134-177126E0BD3F     *EXPLOIT*
|       F8981437-1287-5B69-93F1-657DFB1DCE59    8.1     https://vulners.com/githubexploit/F8981437-
1287-5B69-93F1-657DFB1DCE59     *EXPLOIT*
|       F58A5CB2-2174-586F-9CA9-4C47F8F38B5E    8.1     https://vulners.com/githubexploit/F58A5CB2-
2174-586F-9CA9-4C47F8F38B5E     *EXPLOIT*
|       EFD615F0-8F17-5471-AA83-0F491FD497AF    8.1     https://vulners.com/githubexploit/EFD615F0-
```

Este escaneo reveló numerosas vulnerabilidades potenciales.

El análisis de estas vulnerabilidades confirmó que el servidor estaba ejecutando software con múltiples vulnerabilidades conocidas, lo que aumentaba la superficie de ataque.

Paso 2: Se verificó si el servicio FTP permitía acceso anónimo utilizando dos métodos diferentes para una mayor confirmación:

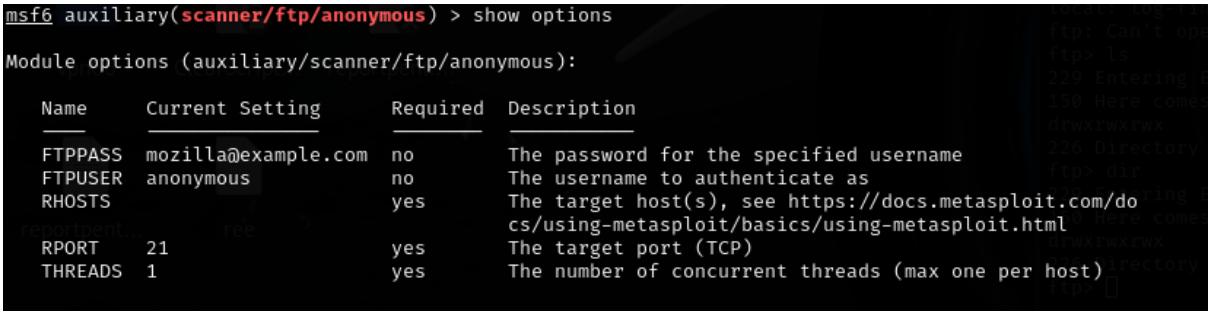
Método 1 - Cliente FTP estándar:

ftp 10.0.2.13

```
Name (10.0.2.13:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Método 2 - Metasploit Framework:

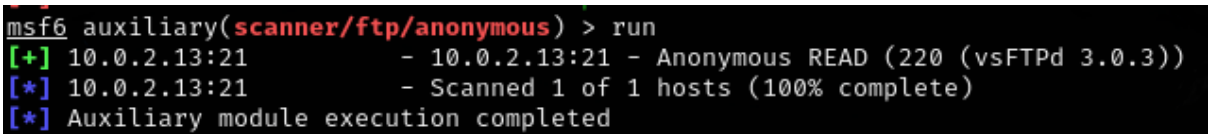
Como parte de la verificación exhaustiva, se utilizó también el módulo auxiliar de Metasploit para detectar acceso FTP anónimo:



msf6 > use auxiliary/scanner/ftp/anonymous

set RHOSTS 10.0.2.13

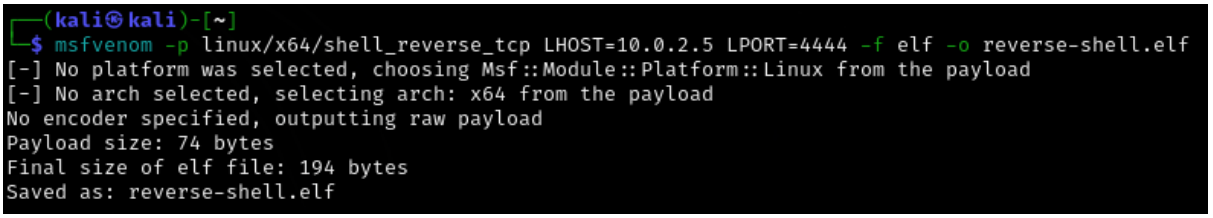
msf6 > run



Esta doble verificación confirma sin lugar a dudas que el servidor está configurado para permitir acceso anónimo, un hallazgo crítico para la seguridad del servidor

Paso 3: Una vez conectado, se realizó un reconocimiento de las carpetas disponibles y sus permisos:

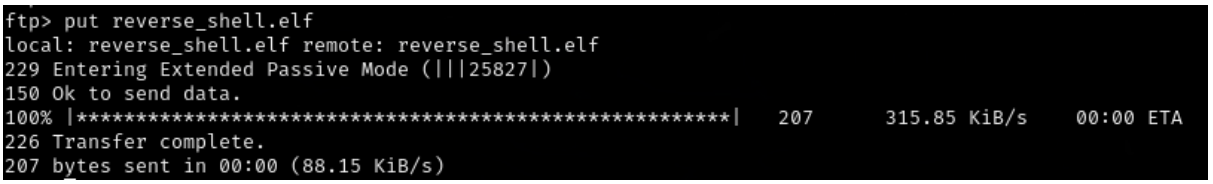
Paso 4: Se procedió a crear una reverse shell utilizando msfvenom para generar un payload ejecutable en sistemas Linux:



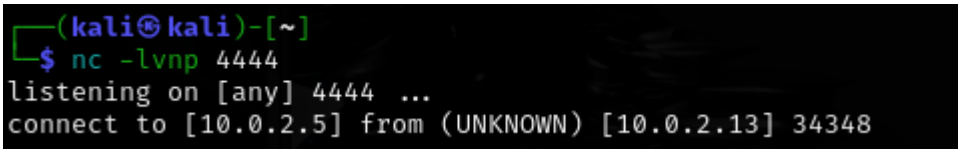
Este comando genera un archivo ejecutable ELF que, al ejecutarse en el servidor objetivo, establecerá una conexión de reverse shell hacia la máquina del

atacante (10.0.2.5) en el puerto 4444. Se eligió específicamente el formato ELF por ser compatible con sistemas Linux como Debian.

Paso 5: A continuación, se cargó la reverse shell en el servidor a través de FTP:



Paso 6: Antes de ejecutar la reverse shell en el servidor objetivo, se configuró un listener netcat en la máquina del atacante para recibir la conexión:



Impacto

Mediante esta vulnerabilidad, un atacante puede:

- **Ejecutar código arbitrario de forma remota** en el servidor con los privilegios del usuario que gestiona el servicio FTP.
- **Obtener acceso inicial al sistema** sin necesidad de credenciales válidas.
- **Comprometer la confidencialidad, integridad y disponibilidad del sistema** al tener capacidad para leer, modificar y potencialmente eliminar información.
- **Establecer persistencia** para mantener el acceso al sistema incluso si se reinicia.
- **Utilizar el servidor comprometido como punto de partida** para atacar otros sistemas en la misma red.

El impacto es especialmente severo debido a la facilidad de explotación (no requiere habilidades avanzadas) y a la ausencia de necesidad de autenticación.

Componentes Afectados

- 10.0.2.13 (Sistema operativo Debian Linux 4.15.x)

Descripción

La escalación de privilegios ocurre cuando un usuario obtiene privilegios superiores a los que debería tener, permitiéndole ejecutar acciones no autorizadas. Esto puede suceder debido a configuraciones incorrectas, permisos excesivos, vulnerabilidades en el sistema operativo o en las aplicaciones instaladas.

En el caso específico evaluado, el servidor Debian ejecutaba una versión vulnerable del kernel Linux (4.15.x) que contenía múltiples vulnerabilidades conocidas que podían ser explotadas para obtener privilegios de root. Estas vulnerabilidades están documentadas en múltiples CVEs con puntuaciones CVSS elevadas.

La escalación de privilegios es especialmente peligrosa cuando se combina con vulnerabilidades de acceso inicial como el FTP anónimo detectado anteriormente, ya que completa un vector de ataque que permite el compromiso total del sistema.

Explotación

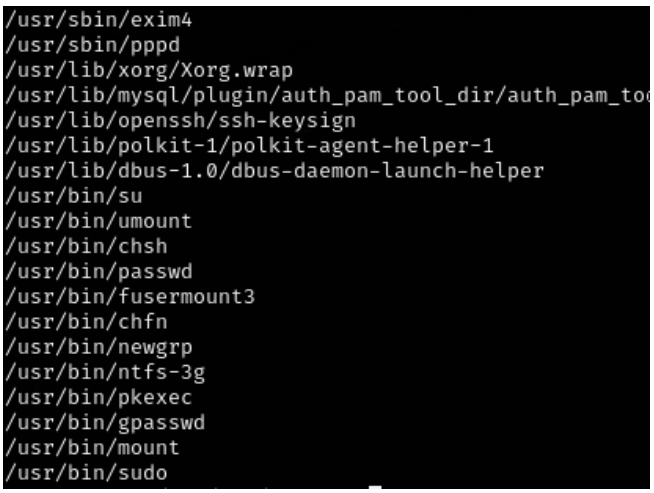
Como continuación del proceso de evaluación para la Fase 2 del proyecto, después de obtener acceso inicial al sistema mediante la vulnerabilidad de FTP anónimo, se procedió a buscar formas de escalar privilegios para obtener control total del servidor.

A continuación, se detalla paso a paso cómo se ha explotado esta vulnerabilidad:

Paso 1: Después de obtener la reverse shell, se realizó una enumeración detallada del sistema para recopilar información sobre el entorno:

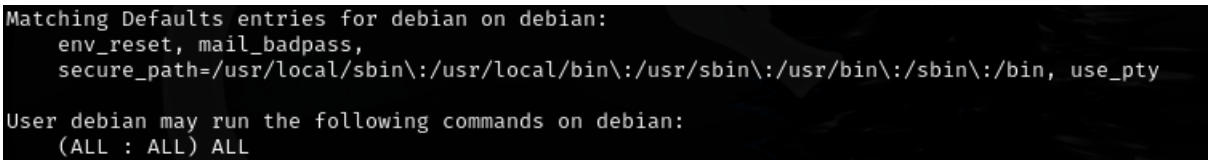
Se verificó la presencia de binarios con bit SUID que podrían ser utilizados para escalar privilegios:

```
find / -perm -u=s -type f 2>/dev/null
```



Especialmente interesantes fueron binarios como `pkexec`, `sudo` y `exim4`, que tienen históricos de vulnerabilidades explotables.

Adicionalmente, se verificaron los permisos de sudo del usuario actual:



Esta información, combinada con el análisis previo de vulnerabilidades del kernel y los binarios SUID, proporcionó múltiples vectores potenciales para la escalación de privilegios. Para la demostración de la Fase 2, se utilizó específicamente la vulnerabilidad del kernel Linux 4.15.x, pero es importante señalar que existían varias rutas alternativas que podrían haberse explotado.

Impacto

A través de esta vulnerabilidad, un atacante puede:

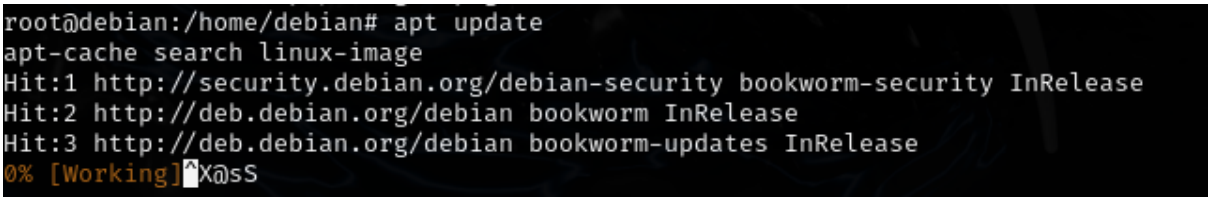
- **Obtener control completo del sistema** con privilegios de administrador (root).
- **Acceder, modificar o eliminar cualquier archivo** en el sistema, independientemente de sus permisos originales.
- **Instalar software malicioso o backdoors persistentes** que sobrevivan a reinicios.
- **Modificar configuraciones críticas del sistema** que podrían afectar su funcionamiento.
- **Acceder a información confidencial** almacenada en el servidor.
- **Utilizar el servidor comprometido como punto de pivote** para atacar otros sistemas en la red interna.
- **Eludir mecanismos de seguridad** que dependen de la separación de privilegios.

La combinación de esta vulnerabilidad con el acceso inicial obtenido a través del FTP anónimo crea un vector de ataque completo que puede ser explotado por atacantes incluso con habilidades técnicas moderadas.

Remediación

Como parte de las acciones correctivas para la Fase 2 del proyecto, se recomienda implementar las siguientes medidas específicas en el servidor de 4Geeks Academy para mitigar las vulnerabilidades de escalación de privilegios:

- **Actualizar inmediatamente el kernel del sistema Debian** a la última versión estable disponible para parchear las vulnerabilidades identificadas:



Tras identificar las versiones disponibles, se recomienda instalar la más reciente compatible:

```
root@debian:/home/debian# apt install linux-image-amd64
```

Alternativamente, se puede instalar directamente la última versión del kernel para la distribución Debian actual:

```
root@debian:/home/debian# apt dist-upgrade
```

si recomienda actualizar a cualquier versión posterior a 5.0, ya que muchas de las vulnerabilidades detectadas afectan específicamente a la serie 4.15.x.

Implementar un programa regular de parcheado de seguridad con el siguiente cronograma:

- Parches críticos: Dentro de las 24 horas de su liberación
- Parches de alta severidad: Dentro de la semana de su liberación
- Parches de media y baja severidad: Mensualmente

Revisar y ajustar permisos de todos los usuarios del sistema siguiendo el principio de privilegio mínimo:

```
grep -r "NOPASSWD" /etc/sudoers /etc/sudoers.d/
```

Conclusiones y Recomendaciones

Análisis Global de las Vulnerabilidades Encontradas

La Fase 2 del proyecto de ciberseguridad para 4Geeks Academy ha revelado vulnerabilidades críticas en el servidor que podrían conducir al compromiso total del sistema. Las principales conclusiones del análisis realizado son:

1. **Vulnerabilidad de acceso FTP anónimo**: La configuración inadecuada del servicio FTP representa una puerta de entrada crítica para atacantes, permitiendo el acceso sin autenticación y la carga de archivos maliciosos. Esta vulnerabilidad constituye un vector inicial de ataque que no requiere credenciales ni conocimientos avanzados.
2. **Múltiples vías para la escalación de privilegios**: El servidor ejecuta una versión obsoleta del kernel Linux (4.15.x) con numerosas vulnerabilidades documentadas. Además, existen varios binarios SUID que podrían ser utilizados para obtener privilegios elevados. Esta combinación de factores facilita que un atacante obtenga control completo del sistema.
3. **Exposición de información sensible**: La configuración actual permite que usuarios no autorizados accedan a datos que podrían revelar detalles sobre la infraestructura y configuración del sistema, facilitando la planificación de ataques más elaborados.
4. **Cadena de ataque completa**: Las vulnerabilidades identificadas pueden encadenarse para formar un vector de ataque completo: acceso inicial a través de FTP anónimo, carga de payload malicioso, ejecución remota de código y escalación de privilegios hasta obtener acceso root.
5. **Ausencia de controles de detección**: Durante las pruebas no se evidenció la presencia de sistemas de detección de intrusiones o monitorización activa que pudieran alertar sobre las actividades maliciosas realizadas.

Estrategia Integral de Remediación

Para abordar las vulnerabilidades identificadas y fortalecer la seguridad del servidor de 4Geeks Academy, se propone una estrategia integral de remediación estructurada en capas:

1. Corrección Inmediata de Vulnerabilidades Críticas

Para el servicio FTP:

- Desactivar completamente el acceso FTP anónimo modificando el archivo de configuración `/etc/vsftpd.conf`
- Si es absolutamente necesario mantener FTP, implementar una autenticación robusta y restringir los permisos de escritura

- Migrar a SFTP como solución más segura a largo plazo

Para el sistema operativo:

- Actualizar el kernel a la última versión estable para corregir las vulnerabilidades de escalación de privilegios
- Aplicar todos los parches de seguridad pendientes mediante `apt update && apt upgrade`
- Revisar y ajustar los permisos de binarios SUID para reducir los vectores potenciales de escalación

2. Implementación de Capas de Defensa Adicionales

Segmentación y control de acceso:

- Implementar reglas de firewall restrictivas que limiten el acceso solo a los servicios necesarios
- Configurar listas de control de acceso basadas en IP para servicios críticos
- Implementar una política obligatoria de control de acceso utilizando AppArmor o SELinux

Monitorización y detección:

- Instalar y configurar un sistema de detección de intrusiones a nivel de host
- Implementar monitorización de logs centralizada
- Configurar alertas automatizadas para comportamientos anómalos o sospechosos

Gestión de vulnerabilidades:

- Establecer un programa regular de escaneo de vulnerabilidades
- Implementar un proceso formal de gestión de parches con tiempos definidos según la criticidad
- Realizar auditorías periódicas de configuración de seguridad

3. Plan de Implementación Secuencial

Para asegurar una implementación eficaz y minimizar el impacto operativo, se recomienda seguir este plan secuencial:

Fase Inmediata (24-48 horas):

1. Desactivar el acceso FTP anónimo
2. Actualizar el kernel y aplicar parches críticos de seguridad
3. Realizar una copia de seguridad completa del sistema
4. Implementar reglas básicas de firewall

Fase de Corto Plazo (1-2 semanas):

1. Migrar de FTP a SFTP
2. Implementar monitorización básica de logs
3. Revisar y ajustar permisos de usuarios y archivos
4. Realizar un escaneo completo de vulnerabilidades

Fase de Medio Plazo (1 mes):

1. Implementar AppArmor o SELinux
2. Configurar un sistema de detección de intrusiones
3. Desarrollar procedimientos de respuesta a incidentes
4. Implementar monitorización avanzada y alertas

Fase de Largo Plazo (3-6 meses):

1. Establecer un programa continuo de gestión de vulnerabilidades
2. Implementar segmentación de red
3. Desarrollar políticas de seguridad formales
4. Realizar pruebas de penetración periódicas

Lecciones Aprendidas y Consideraciones Futuras

El análisis realizado durante la Fase 2 del proyecto ha permitido extraer varias lecciones importantes que deberían guiar las estrategias de seguridad futuras.

1. **La importancia de la configuración segura**: Una configuración incorrecta del servicio FTP ha sido el punto de entrada inicial para el compromiso del sistema. Esto subraya la necesidad de implementar y mantener configuraciones seguras para todos los servicios.
2. **El valor de mantener sistemas actualizados**: La presencia de un kernel desactualizado con múltiples vulnerabilidades conocidas facilitó la escalación de privilegios. Un programa efectivo de gestión de parches podría haber mitigado este riesgo.
3. **La necesidad de adoptar el principio de mínimo privilegio**: La existencia de numerosos binarios SUID y permisos excesivos amplió la superficie de ataque. Aplicar el principio de mínimo privilegio reduciría significativamente las posibilidades de escalación.
4. **La defensa en profundidad como estrategia crucial**: Un único control de seguridad fallido (en este caso, la configuración del FTP) no debería comprometer todo el sistema. La implementación de múltiples capas de defensa habría dificultado significativamente el avance del ataque.

5. **El rol de la monitorización proactiva**: Un sistema efectivo de monitorización habría detectado las actividades sospechosas durante las fases iniciales del ataque, permitiendo una respuesta más rápida.

De cara al futuro, se recomienda que 4Geeks Academy considere implementar un programa integral de seguridad que incluya:

- Establecer un ciclo de vida de desarrollo seguro (S-SDLC) para aplicaciones
- Realizar evaluaciones de seguridad periódicas, incluyendo pruebas de penetración
- Desarrollar un programa de concienciación en seguridad para el personal
- Implementar un proceso formal de gestión de riesgos de seguridad
- Considerar la adopción de un framework de seguridad reconocido como NIST CSF o ISO 27001

Impacto en el Negocio y Conclusión Final

Las vulnerabilidades identificadas en esta fase del proyecto representan un riesgo significativo para 4Geeks Academy. Un compromiso exitoso del servidor podría resultar en:

- **Pérdida de confidencialidad** de datos sensibles de la academia y sus estudiantes
- **Interrupción de servicios educativos** si los sistemas se vuelven inaccesibles
- **Daño reputacional** si se hace pública una brecha de seguridad
- **Posibles sanciones regulatorias** dependiendo de las leyes aplicables de protección de datos
- **Costos financieros** asociados con la respuesta a incidentes y recuperación

La implementación de las recomendaciones detalladas en este informe no solo mitigará las vulnerabilidades específicas identificadas, sino que también fortalecerá la postura general de seguridad de la organización. Invertir en seguridad debe verse como una protección del valor del negocio y de la confianza depositada por estudiantes y colaboradores en la academia.

La seguridad es un proceso continuo, no un estado final. Por tanto, se recomienda que 4Geeks adopte un enfoque proactivo y evolutivo hacia la ciberseguridad, adaptándose constantemente a las nuevas amenazas y desafíos del entorno digital.