Boletín de Seguridad Informática

Análisis de Vulnerabilidad CVE-2024-6387

Ejecución Remota de Código (RCE)

Descripción breve

Este boletín se enfoca en la vulnerabilidad CVE-2024-6387, el cual consiste en una regresión de seguridad de la vulnerabilidad CVE-2006-5051 en OpenSSH. Esta vulnerabilidad puede causar denegación de servicio o incluso ejecución de código remoto como root.

M.I. Adriana Cruz García
Ing. Andrés Martínez López
Ing. César Alejandro Varela Cruz

csi.incidentes@unam.mx

Evaluado por:
Ing. Julio César Roldán Elorza

Avalado por:
M. en C. Carlos R. Tlahuel Pérez

Contenido

Resumen Ejecutivo

La vulnerabilidad CVE 2024-6387 es una regresión de seguridad (CVE-2006-5051) en el servicio de OpenSSH. Esta vulnerabilidad ocurre cuando dos o más procesos o hilos intentan acceder o modificar un recurso compartido en paralelo sin una sincronización adecuada, a esto se le conoce como *Condición de carrera*. Esta condición se da en el procesamiento de señales del sistema, situación que puede provocar que los demonios asociados al servicio manejen señales inseguras.

Al tener esta vulnerabilidad y con la configuración predeterminada de *sshd*[1], si un usuario no se autentica dentro de un tiempo determinado *(LoginGraceTime*[2]), automáticamente se llama al controlador *SIGALRM*[3] de *sshd* de manera asíncrona, llamando a otras funciones inseguras. De esta forma, si un atacante remoto logra interactuar con el proceso de autenticación, podría interrumpir o alterar el comportamiento de *sshd* para enviar señales arbitrarias a dicho proceso, alterando su funcionamiento normal, incluso rechazando las conexiones válidas, permitiendo el acceso no autorizado al sistema, causando una posible ejecución remota de código no autenticado con privilegios root o la caída del servicio a través de una denegación de servicio si se agotan todas las conexiones.

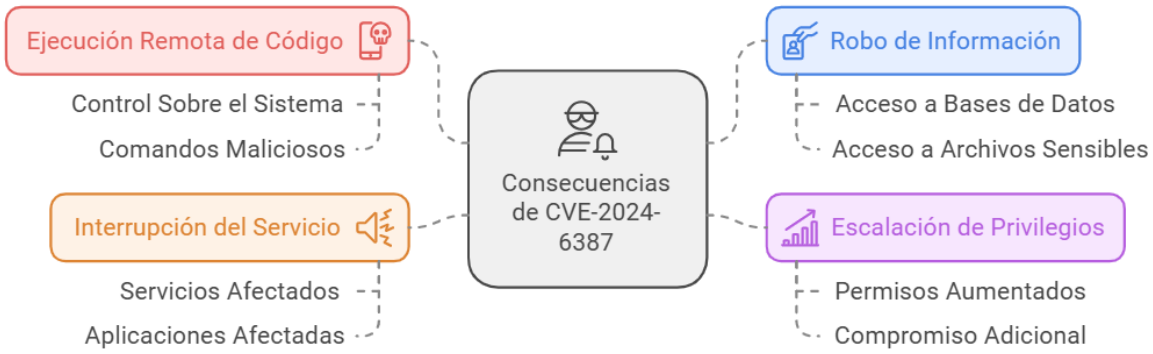

Ilustración 1. Consecuencias de la vulnerabilidad CVE 2024-6387

Por lo tanto, una inadecuada sincronización en los procesos instanciados por *sshd* para manejar las señales, permite que el proceso entre en un estado inconsistente o vulnerable si recibe una señal en un momento inesperado. Así, el ataque se realiza sin necesidad de tener credenciales válidas, lo que aumenta la gravedad de la vulnerabilidad.

[1] Secure Shell Daemon es el servicio o "demonio" que permite la conexión a través de Secure Shell en sistemas Unix y Linux.
[2] Parámetro del servidor SSH que especifica el tiempo permitido para una autenticación exitosa.
[3] Es una señal en sistemas operativos tipo Unix que se utiliza para indicar que ha expirado un temporizador o que ha ocurrido un evento relacionado con el tiempo.

Prevención:

- Configuración del parámetro *LoginGraceTime*
- Uso de herramientas de prevención de intrusos
- Configuración de ASLR (Address Space Layout Randomization)
- Configuración de NX (No eXecute)
- Implementar una medida de bloqueo de IP basado en intentos fallidos de ssh

Impacto

La vulnerabilidad CVE 2024-6387 se considera de alto impacto con una severidad de 8.1, afectando ampliamente la Confidencialidad, Integridad y Disponibilidad en el sistema, representado en el siguiente vector dentro del Sistema de Puntuación de Vulnerabilidades Comunes (CVSS): CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

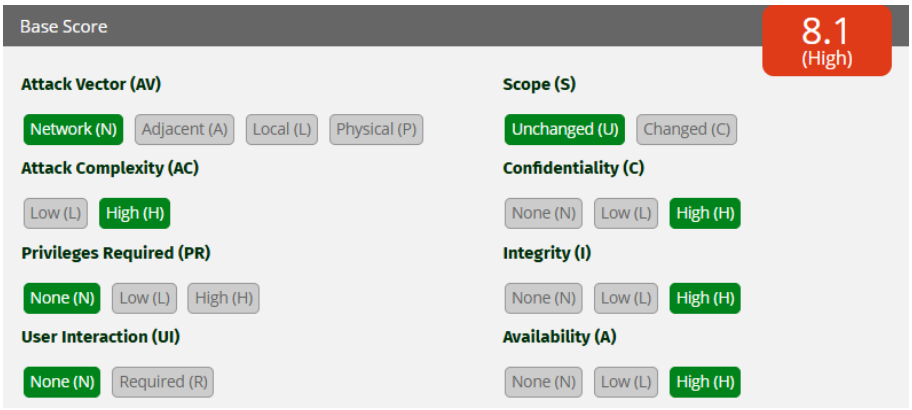

Ilustración 2. Impacto CVE 2024-6387 (First, 2024)

La complejidad para explotar está vulnerabilidad se considera alta debido a que la ventana de oportunidad que tiene para inyectar un *payload*[4] y corromper el servicio, es mínima, así como para lograr una ejecución de código remota (RCE). Además de que un intento de ataque no puede ejecutarse al instante posterior a otro, la condición de carrera solo se presenta una vez transcurrido el *LoginGraceTime*, que aunado a conexiones SSH ilimitadas cada 2 o 5 minutos, hace que la ejecución del ataque pueda ser exitosa a pesar de demorar semanas.

De forma complementaria, hoy en día existen diversas pruebas de concepto en múltiples foros y repositorios de internet, teniendo una probabilidad de explotar esta vulnerabilidad con una puntuación de 2.2, que, a pesar de su alta complejidad, puede causar un gran impacto en las tecnologías que presentan esta brecha de seguridad vigente.

[4] Parte del código del malware que realiza la acción maliciosa en el sistema

Base Score	Base Severity	CVSS Vector	Exploitability Score	Impact Score	Score Source	First Seen
8.1	HIGH	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	N/A	N/A	Cisco:cisco-sa-openssh-rce-2024	2024-07-04
8.1	HIGH	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	N/A	N/A	Red Hat, Inc.	2024-07-01
8.1	HIGH	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	2.2	5.9	NIST	2024-07-05
8.1	HIGH	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	2.2	5.9	Red Hat, Inc.	2024-07-01

Ilustración 3. Evaluación de la explotación (cvedetails, 2024)

Descripción de la vulnerabilidad

La vulnerabilidad con el identificador CVE-2024-6387, denominada coloquialmente "RegreSSHion", es un derivado de una vulnerabilidad anterior con el identificador CVE-2006-5051, la cual fue mitigada en septiembre del 2006 en la versión 4.4p1 de OpenSSH, sin embargo, al ser una tecnología de software libre que se encuentra en constante cambio por una comunidad muy amplia, el parche para la vulnerabilidad fue eliminado en el commit 752250c[5]realizado en octubre del 2020 el cual corresponde a la versión 8.5p1, por lo que la vulnerabilidad volvió a presentarse y fue parchada nuevamente en el commit 81c1099[6] correspondiente a la versión 9.8p1, por lo que las versiones vulnerables son:

- OpenSSH < 4.4p1 (CVE-2006-5051)
- 8.5p1 <= OpenSSH < 9.8p1 (CVE-2024-6387)

Ilustración 4. Commit de repositorio GitHub en el que se eliminó el parche del CVE-2006-5051 (GitHub, 2024)

[5] https://github.com/openssh/openssh-portable/commit/752250caabda3dd24635503c4cd689b32a650794
[6] https://github.com/openssh/openssh-portable/commit/81c1099d22b81ebfd20a334ce986c4f753b0db29

Esta vulnerabilidad se puede explotar de forma remota en sistemas Linux basados en *glibc*[7], donde *syslog()*[8] llama a funciones *async-signal-unsafe*[9](por ejemplo, *malloc()*[10] y *free()*[11]), lo cual puede ocasionar la ejecución de código remoto no autenticado con privilegios de *root*, producido por la modificación al código de *sshd*.

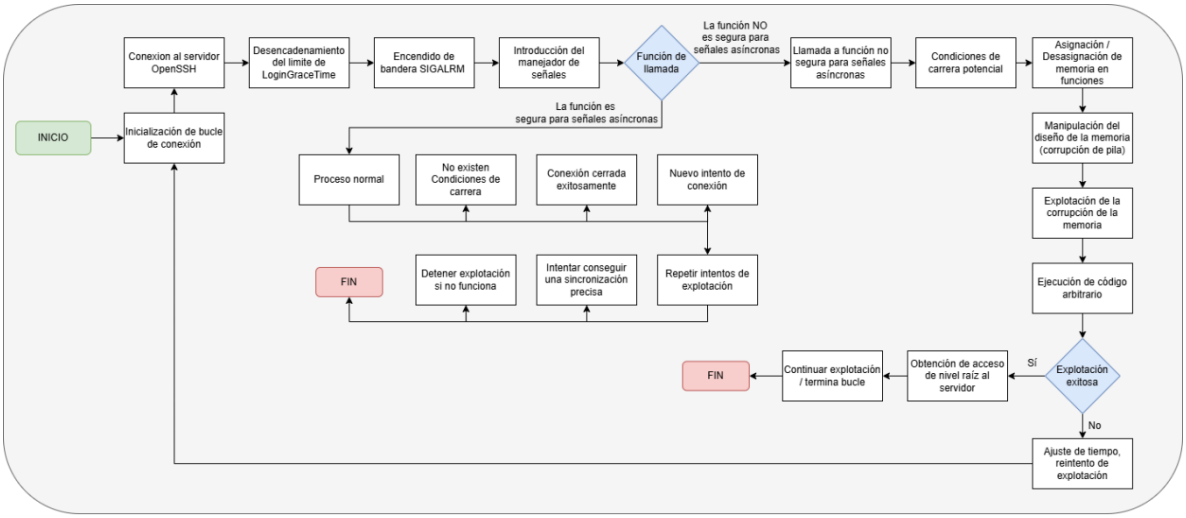

Ilustración 5. Vulnerabilidad CVE 2024-6409 (Trendmicro, 2024)

Esta función asíncrona (syslog) se ejecuta posterior al "*LoginGraceTime*" es decir, al no realizar un intento de inicio de sesión durante 2 minutos (120 segundos), sin embargo, en algunas versiones anteriores de OpenSSH esta función podría ejecutarse hasta los 5 minutos (600 segundos), cuyo propósito es terminar la comunicación con el cliente que intenta conectarse para evitar que las solicitudes se almacenen indefinidamente y puedan representar un problema para el servidor.

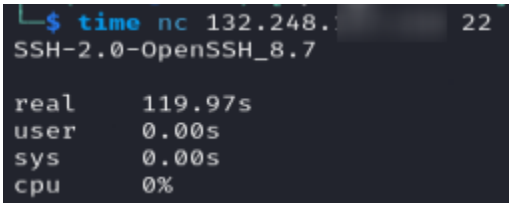

Ilustración 6. Tiempo de ejecución.

[7] Conjunto de bibliotecas básicas para los sistemas GNU/Linux y GNU
[8] Protocolo de registro de eventos
[9] Función del controlador de señales
[10] Asignación dinámica de memoria
[11] Liberación dinámica de memoria

Prevención

Esta vulnerabilidad no solo depende de la tecnología OpenSSH, ya que el sistema operativo sobre el que esté instalado juega un papel muy relevante, esto se debe a que diversos sistemas modernos cuentan con medidas de seguridad, si bien no es posible garantizar la seguridad total de un sistema, es posible prevenir y reducir la posibilidad de que un sistema sea explotado, para esto se ofrecen las siguientes recomendaciones:

- Mantener el software actualizado a su última versión estable. Ya que estas versiones normalmente cuentan con todos los parches de seguridad y son menos propensos a bugs de lo que lo sería una versión alfa, beta, etc.
- Implementar una configuración segura del servidor, para esto se puede hacer uso de guías como las CIS Benchmarks.
- Configurar el servidor sshd ante una denegación de servicio
    - Autenticarse como *root* [12]e ingrese a la ruta: */etc/ssh/sshd_config*
    - Modificar o en su defecto agregar la configuración del parámetro:
        - *LoginGraceTime 0*
    - Guardar y cerrar el archivo
    - Reiniciar el demonio sshd
        - *Systemctl restart sshd.service*

  Esta configuración permitirá deshabilitar el llamado automático del controlador SIGALRM de sshd de manera asíncrona, y las demás señales derivadas de este.
- Uso de herramientas de prevención de intrusos
  Se recomienda utilizar herramientas como "*fail2ban*" [13]en conjunto con un *firewall* [14]
- Habilitar ASLR (Address Space Layout Randomization)
  Esta técnica asigna a los componentes críticos del sistema una dirección de memoria diferente y de forma aleatoria a los ejecutables o bibliotecas utilizadas. De esta manera es más complicado para un atacante utilizar *exploits*[15] que dependan de direcciones fijas o permanentes, sin embargo, a pesar de que algunos sistemas cuentan con esta medida de seguridad, siguen teniendo direcciones preestablecidas para bibliotecas del sistema, por lo que algunos exploits siguen utilizando direcciones como la 0xb7200000 y 0xb7400000, las cuales son direcciones en las que comúnmente los sistemas operativos de 32bits almacenan la biblioteca glibc.

```
44      // Possible glibc base addresses (for ASLR bypass)
45      uint64_t GLIBC_BASES[] = { 0xb7200000, 0xb7400000 };
46      int NUM_GLIBC_BASES = sizeof (GLIBC_BASES) / sizeof (GLIBC_BASES[0]);
47
```

Ilustración 7. Muestra de exploit público (GitHub, 2024))

[12] Permisos de administrador
[13] Aplicación escrita en Python para la prevención de intrusos en un sistema
[14] Sistema de seguridad que filtra y controla el tráfico de red
[15] Software diseñado para aprovechar un fallo en un sistema informático

Para verificar si esta protección se encuentra habilitada, en la mayoría de los sistemas operativos Linux se puede revisar el contenido del archivo /proc/sys/kernel/randomize_va_space cuyos valores pueden ser los siguientes:

- 0: Deshabilitado.
- 1: Habilitado parcialmente (solo para algunos componentes).
- 2: Habilitado completamente (recomendado).

- Habilitar NX (No eXecute)

  NX, también conocido como **DEP** (Data Execution Prevention) es una característica de seguridad que etiqueta ciertas áreas de la memoria como "No ejecutables", lo cual evita que se ejecute código almacenado en espacios de memoria con este etiquetado. En ocasiones, dependiendo de lo robusto que sea un programa, esto podría detener la ejecución de código malicioso, sin embargo, puede causar una denegación de servicio ya que al intentar ejecutar el código desde un espacio de memoria no ejecutable los programas generan un error y en ocasiones pueden detenerse.

  En Linux los pasos para verificar si esta protección se encuentra activa puede variar entre distribuciones o incluso entre versiones de las distribuciones. Además, no necesariamente las versiones más recientes de una distribución de Linux son las que cuentan con esta prevención de ejecución de datos activada por defecto, por lo que se deben revisar las características o documentación de la distribución, así como verificar si el procesador del equipo soporta esta prevención. La mayoría de las distribuciones de Linux cuentan con un documento o sección dentro de su página web en la que describen las características de las últimas versiones del sistema operativo, por ejemplo:

  - Debian: https://wiki.debian.org/Security/Features
  - Ubuntu: https://wiki.ubuntu.com/Security/Features
  - Fedora: https://fedoraproject.org/wiki/Security_Features

Estas medidas de seguridad pueden ayudar a prevenir o reducir el impacto de esta y muchas otras vulnerabilidades existentes por lo que de no contar con un sistema operativo que las tenga habilitadas por defecto se recomienda implementarlas.

Conclusión

En los últimos años, la Universidad Nacional Autónoma de México ha tenido un crecimiento considerable en el desarrollo tecnológico, incluyendo modificaciones y creación de nuevos sistemas universitarios críticos, su importancia radica en la gestión de grandes volúmenes de información, tanto académica como administrativa, y en el apoyo a los procesos clave que permiten el funcionamiento diario de la universidad.

Dado que la universidad maneja una amplia diversidad de sistemas operativos, y de diversos servicios para su administración como OpenSSH, es imperativo adoptar las medidas adecuadas para la prevención de incidentes de seguridad.

El estudio antes realizado sobre la vulnerabilidad CVE-2024-6387 que expone riesgos asociados a errores de sincronización y condiciones de carrera en servicios críticos como SSH, subraya la necesidad de contar con un enfoque de seguridad integral, que no solo aborde los riesgos

evidentes, sino también aquellos que pueden ser explotados de manera sutil pero eficaz. A pesar de que la probabilidad de explotación puede parecer baja, se destaca el hecho de que es derivada de una vulnerabilidad detectada anteriormente, resaltando la importancia de actualizar y reforzar constantemente las medidas de protección; ya que incluso errores aparentemente menores pueden ser aprovechados por atacantes para obtener acceso no autorizado y comprometer la infraestructura tecnológica, afectando la operación y seguridad de la universidad.

Referencias

- Jump over ASLR: Attacking branch predictors to bypass ASLR. (19 de octubre de 2016). Obtenido de IEEE Xplore: https://ieeexplore.ieee.org/abstract/document/7783743
- Kaspersky Team. (5 de Julio de 2024). Attack on cybersecurity researchers: pseudo-exploit for regreSSHion. Obtenido de Kaspersky: https://www.kaspersky.com/blog/cve-2024-6387-regresshion-researcher-attack/51646/
- Microsoft. (12 de junio de 2023). Protección DEP/NX. Obtenido de Microsoft Ignite: https://learn.microsoft.com/es-es/windows/win32/win7appqual/dep-nx-protection
- OffSec Team. (8 de Julio de 2024). RegreSSHion exploit, CVE-2024-6387: A Write-Up. Obtenido de OffSec: https://www.offsec.com/blog/regresshion-exploit-cve-2024-6387/
- P., M. (2 de Julio de 2024). regreSSHion. Obtenido de github: https://github.com/xonoxitron/regreSSHion
- Qualys. (1 de Julio de 2024). regreSSHion: Remote Unauthenticated Code Execution Vulnerability in OpenSSH server. Obtenido de https://blog.qualys.com/vulnerabilities-threat-research/2024/07/01/regresshion-remote-unauthenticated-code-execution-vulnerability-in-openssh-server
- Red Hat, Inc. (1 de Julio de 2024). CVE-2024-6387 Detail. Obtenido de https://nvd.nist.gov/vuln/detail/cve-2024-6387
- CVE website. (1 de julio de 2024). CVE-2024-6387. Obtenido de https://www.cve.org/CVERecord?id=CVE-2024-6387
- Benatto, Marco. (1 de julio de 2024). Bug 2294604 (CVE-2024-6387, regreSSHion) - CVE-2024-6387 openssh: regreSSHion - race condition in SSH allows RCE/DoS. Obtenido de https://bugzilla.redhat.com/show_bug.cgi?id=2294604
- Shastri, Jagir (17 de julio de 2024). The Potential Impact of the OpenSSH Vulnerabilities CVE-2024–6387 and CVE-2024-6409. Obtenido de https://www.trendmicro.com/en_us/research/24/g/cve-2024-6387-and-cve-2024-6409.html