



Kiddie No More: Learning from your exploits

Ruben Ventura Piña [tr3w]
BugCon 2009

trew@localh0st ~ \$ whoami

- Ruben Ventura Piña [tr3w]
- Coordinador del grupo de investigación de seguridad informática del ITESM Campus Aguascalientes.
- Application Security Assessment at Proseg Information Security
- Investigación, análisis y escritura de exploits

Objetivo

- Proporcionar al asistente la metodología y motivación de desarrollar las habilidades necesarias para que pueda utilizar un exploit o PoC como herramienta de aprendizaje y tenga una comprensión amplia del mismo.

Agenda

- La seguridad informática en el mundo actual
 - El papel de los *Script Kiddies*
 - El papel de los “expertos en seguridad”
 - El papel de los *PoC* y *exploits*
- Caso práctico de la vida real – *Local root exploit*
 - Comunicación entre *userspace* y *kernel space*
 - *Netlink sockets*
 - Estructura y control de mensajes UEVENT
 - Dispositivos de bloque y funcionamiento de udev
 - Estructura de archivos en EXT2/3/4

La Seguridad en el mundo actual

- La información e internet como herramienta base
- Amenaza diaria: BlackHats, Script Kiddies
- Pérdida de información o servicios = \$--
- Nueva inversión en protección y seguridad
- Surgen soluciones: empresas/productos de consultoría/auditoría, “expertos en seguridad”, WhiteHats.

top
+frenzy
a collaboration of imaginations
throwing
everything I have right at you
and you only thought that you knew
looking at "it" as an enemy
do you know who you are?
following
you must decide what you believe
making the choice
is all that matters

ScriptKiddies en la escena

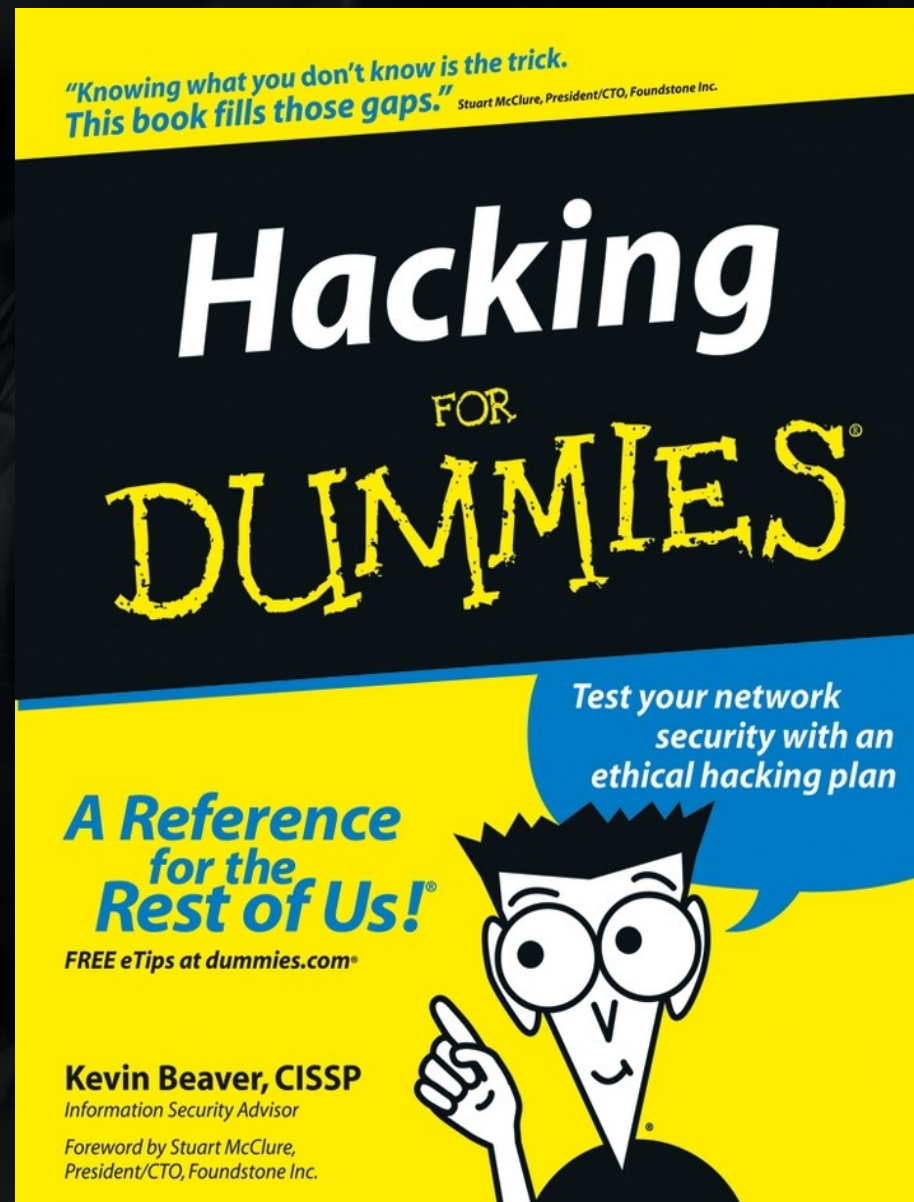


ScriptKiddies en la escena



top
+frenzy
a collaboration of imaginations
throwing
and you only thought that you knew
ing I have right at you
forgotten moment
following
you must decide what you believe
making the choice
may be
is all that matters

ScriptKiddies en la escena



top
+frenzy
a collaboration of imaginations
everything is a false image
you must decide what you believe
making the choice
is all that matters
may be
do you know who you are?
possibly
forgotten moment
ing I have right at you
and you only thought that you knew
following
throwing
V1

ScriptKiddies on the scene

- No entienden los sistemas ni seguridad
- Se alimentan de exploits públicos y del *full disclosure*
- Suelen ser destructivos
- Están al pendiente de nuevos *bugs* y *exploits*
- Pueden ser una amenaza = riesgo para la integridad de una empresa

WhiteHats & Security Professionals on the scene

- Son pocos los que de verdad dan protección
- ***Security course + 10 book < lifetime hacker***
- Muchos no tienen, o no se dan el tiempo de mantenerse actualizados con técnicas de intrusión.
- ***Nmap + nessus + metasploit != PenTesting***
- ***El falso sentido de la seguridad.***

PoCs and exploits in the scene

- *“Know your enemy and know yourself”* - Sun Tzu
- Proveen información necesaria para asegurar sistemas y redes.
- Un *proof-of-concept* o un *exploit* determina con certeza vulnerabilidades = análisis más exactos
- Ha habido polémica acerca de la responsabilidad y ética que implican los *working exploits* y el *full disclosure*.

PoCs and exploits in the scene

- Surge el *responsible disclosure*
 - Proporciona muy poca información y no hay PoC
- El código, estructura y metodología de los *exploits* son cosas complejas.
- Entender las vulnerabilidades es algo esencial para realizar análisis precisos.
 - El caso de Open0wn
- Entender un fallo de seguridad implica tener un conocimiento fuerte y amplio en diversas áreas

Learning from expl0its to understand them

- Independientemente de quien seas, entender un exploit es importante.
- Te garantiza certeza en tus análisis
- Te permite identificar posibles blancos de atacantes y priorizarlos
- Te da el criterio para poder juzgar el servicio que te da un producto o una persona.
- Los chicos malos saben hacer todo esto

Ejemplo Práctico: Escalando a root por medio de udev

Como escribir un *root exploit* y no darse por
vencido en el intento.

Why udev?

- Al leer de la vulnerabilidad quedé intrigado por su funcionamiento.
- Comencé a investigar y cuando al fin tuve lo necesario para entender el fallo, quedé fascinado con la naturaleza de la vulnerabilidad y el método de explotación.
- Para entenderlo tuve que estudiar muchos temas y tecnologías. Se necesita un conocimiento amplio para lograr explotarlo.

Why udev?

- Es un *local root exploit* – No les dan la importancia que merecen → Mayor impacto
- Cada vez se implementa una mayor cantidad de sistemas complejos de defensa como el *Mandatory Access Control*, y la Seguridad por Virtualización.
- Por lo que los fallos de escalación de privilegios se vuelven más importantes.

@ the pwnie awards

- La vulnerabilidad de udev fue nomida en los *pwnie awards* de *BlackHat* en la categoría “*Best Privilege Escalation Bug*”
- “Award to the person who discovered and/or exploited the most technically sophisticated and interesting privilege escalation vulnerability.”

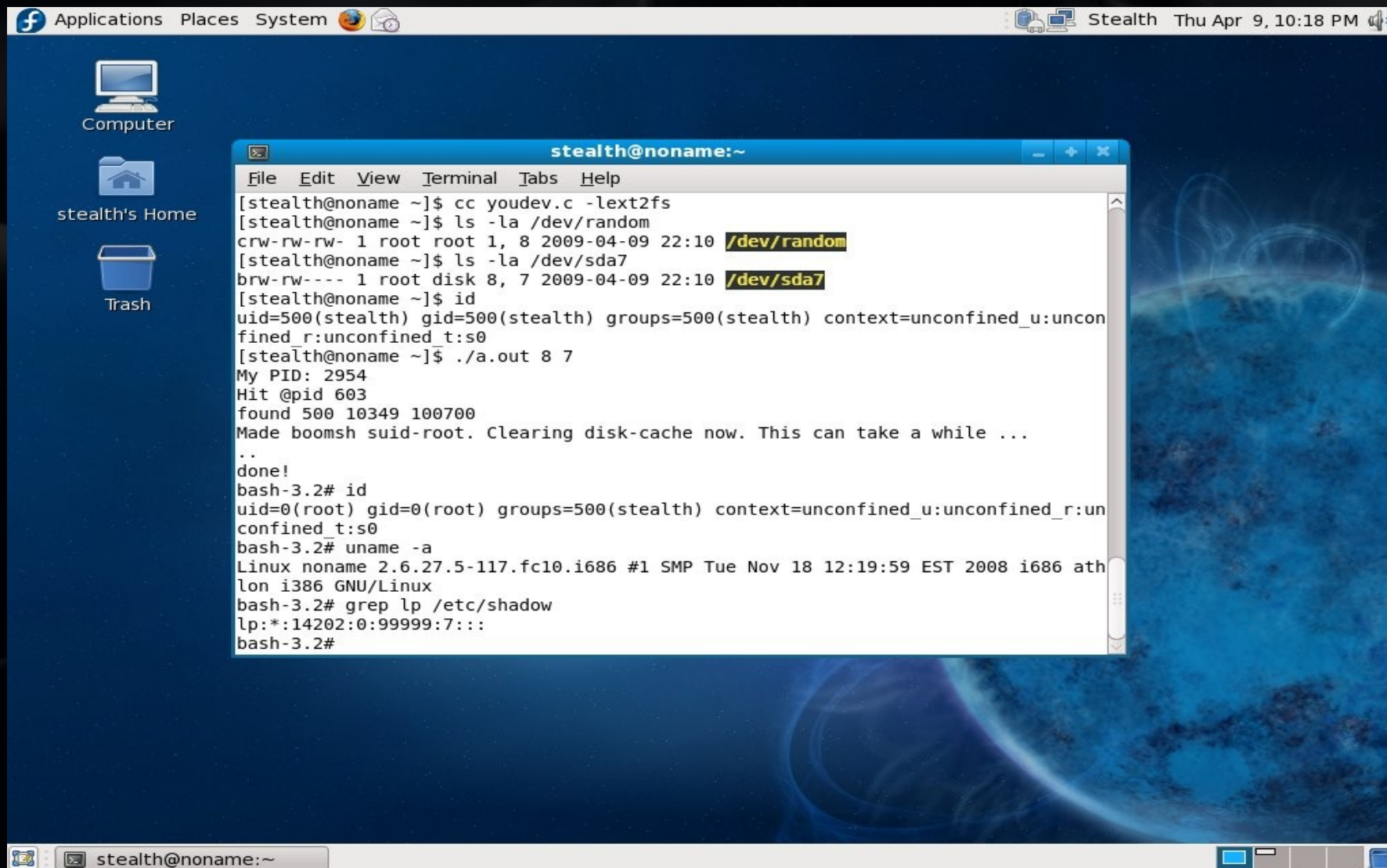
Vulnerability History

- Descubierta por Sebastian Krahmer (*CVE-2009-1185*)
- Sebastian hace *responsible disclosure* en su blog [*16-april-2009*]
- *BugTraq* y otras BD de vulnerabilidades la anuncian [*17-abril-2009*]
- *Gentoo Linux* libera parche [*18-abril-2009*]
- *Milw0rm* publica el primer exploit [*20-abril-2009*]

My History

- Cuando me enteré del *bug* sólo había un par de *advisories* y ningun *PoC*.
- Después de realizar investigación exhaustiva, la noche del 17 de abril ya tenía un *exploit* funcional.
- Sin embargo ya había *exploits* funcionales ronando en internet el mismo día que Sebastian publicó el *bug*.

The Start Line - 16-04-2009



The screenshot shows a Linux desktop with a blue background. On the left, there are icons for 'Computer', 'stealth's Home', and 'Trash'. The top panel includes a menu bar with 'Applications', 'Places', and 'System', and a status bar on the right showing 'Stealth Thu Apr 9, 10:18 PM'. A terminal window titled 'stealth@noname:~' is open, displaying the following commands and output:

```
File Edit View Terminal Tabs Help
[stealth@noname ~]$ cc youdev.c -lxt2fs
[stealth@noname ~]$ ls -la /dev/random
crw-rw-rw- 1 root root 1, 8 2009-04-09 22:10 /dev/random
[stealth@noname ~]$ ls -la /dev/sda7
brw-rw---- 1 root disk 8, 7 2009-04-09 22:10 /dev/sda7
[stealth@noname ~]$ id
uid=500(stealth) gid=500(stealth) groups=500(stealth) context=unconfined_u:unconfined_r:unconfined_t:s0
[stealth@noname ~]$ ./a.out 8 7
My PID: 2954
Hit @pid 603
found 500 10349 100700
Made boomsh suid-root. Clearing disk-cache now. This can take a while ...
..
done!
bash-3.2# id
uid=0(root) gid=0(root) groups=500(stealth) context=unconfined_u:unconfined_r:unconfined_t:s0
bash-3.2# uname -a
Linux noname 2.6.27.5-117.fc10.i686 #1 SMP Tue Nov 18 12:19:59 EST 2008 i686 athlon i386 GNU/Linux
bash-3.2# grep lp /etc/shadow
lp:!:14202:0:99999:7:::
bash-3.2#
```

Screenshot by Sebsatian Krahmer <http://c-skills.blogspot.com/>

The Start Line - 16-04-2009

- *“The first problem appears since the origin of KOBJECT_UEVENT messages are not verified, so any user can spoof messages that udevd takes as granted from kernel. This allows some trickery to hijack any blockdevice through its minor and major numbers and give it permission 0666. The rest is code.”*

– From Sebastian Krahmer's Blog

Let's start with the basics

- ¿Qué es udev? ¿Cómo funciona?
- *“udev is the device manager for the Linux 2.6 kernel series. Primarily, it manages device nodes in /dev. It is the successor of devfs and hotplug, which means that it handles the /dev directory and all user space actions when adding/removing devices. The latest versions of udev depend on the latest version of the uevent interface of the Linux kernel.” - Wikipedia*

User-space and Kernel-space

- En Linux, sólo el código más esencial y crítico está integrado en el *kernel*
- Todos los demás elementos (aquello que puede ser identificado con un PID) corren en el user-space.
- Son dos regiones de memorias separadas, con distintos privilegios.
- Necesitan comunicación

Inter-Process Communication

- Los IPCs son métodos que existen entre el userspace y kernelspace
 - *System Calls*
 - *ioctl (Input/Output Controls)*
 - */proc filesystem*
 - *Netlink Sockets*



top
everything is a false image
you must decide what you believe
making the choice
is all that matters
following
may be
a collaboration of imaginations
you know
tendency

Netlink Sockets

- Un IPC usado para transferir información entre el kernel y procesos en el user-space.
- Crea un enlace de comunicación de 2 canales
- Se usa el API estándar para los procesos,
- Los Netlink sockets son usados para mandar mensajes UEVENT (usados por udev).

Making Netlink Sockets

- El *address family* se llama ***AF_NETLINK***
- El *protocol family* se llama ***PF_NETLINK***
- El tipo de socket puede ser ***SOCK_RAW*** o ***SOCK_DGRAM***
- Hay varios tipos de protocolo para realizar diferentes funciones
 - Definidos en `/usr/include/linux/netlink.h`

/usr/include/linux/netlink.h

```
#define NETLINK_ROUTE      0    /* Routing/device hook                               */
#define NETLINK_USERSOCK   2    /* Reserved for user mode socket protocols          */
#define NETLINK_FIREWALL   3    /* Firewalling hook                                   */
#define NETLINK_INET_DIAG  4    /* INET socket monitoring                             */
#define NETLINK_SELINUX    7    /* SELinux event notifications                       */
#define NETLINK_ISCSI      8    /* Open-iSCSI */
#define NETLINK_AUDIT      9    /* auditing */
#define NETLINK_FIB_LOOKUP 10
#define NETLINK_CONNECTOR  11
#define NETLINK_NETFILTER  12   /* netfilter subsystem */
#define NETLINK_IP6_FW     13
#define NETLINK_DNRTMSG    14   /* DECnet routing messages */
#define NETLINK_KOBJECT_UEVENT 15 /* Kernel messages to userspace */
#define NETLINK_GENERIC    16
#define NETLINK_SCSITRANSPORT 18 /* SCSI Transports */
```

Netlink Socket API

- TCP/IP Socket:

```
sockfd = socket(PF_INET, SOCK_STREAM, 0)
```

- Netlink Socket:

```
sockfd = socket(PF_NETLINK, SOCK_DGRAM,  
                NETLINK_KOBJECT_UEVENT)
```

- Los APIs estándar para sockets sirven igual
(socket(), sendmsg(), recvmsg(), close())

Sending Netlink messages

```
sockfd = socket()  
sendmsg(sockfd, &msg, 0)
```

Struct msghdr msg;

msg_name *	(void*)&target_addr
msg_namelen	sizeof(target_addr)
msg_iov *	(void*)&iovec
msg_iovlen	1

Struct iovec iovec;

iov_base *	(void*)&payload
iov_len	NLMSG_SPACE()

Struct sockaddr_nl target_addr;

nl_family	AF_NETLINK
nl_pid	destpid (udevd pid)
nl_groups	0

```
char *payload = malloc (NLMSG_SPACE())
```

UDEV MSG

```
add@uritonto  
DEVPATH=/dev/random  
wawawawa....
```

top
vision of imaginations
everything is a false image
you must decide what you believe
following
making the choice
is all that matters

Back to udev

- Ahora sabemos como se comunica el kernel con udev.
- No sabemos como funciona udev, ni como son los mensajes que recibe del kernel.
- Nos estamos acercando al final, sólomente tenemos que afilar más las preguntas.

top
+frenzy
a collaboration of imaginations
throwing
and you only thought that you knew
everything I have right at you
looking at the forgotten moment
following
you must decide what you believe
making the choice
is all that matters

How does it work?

- Al conectar un dispositivo a tu máquina, el kernel lo detecta.
- Debe ser agregado a /dev para poder usarlo.
- udev no sabe que se agregó un dispositivo, así que el kernel se lo comunica con un mensaje indicándole como se llama el device y otras características.
- udev al recibir el mensaje obedece y actúa

~# udevadm monitor

- Permite ver los mensajes que recibe udev del kernel y lo que le responde.

```
$ man udev
```

- Mensaje para agregar un dispositivo de bloque:

```
add @uritonto  
DEVPATH=/dev/sda3  
MAJOR=8  
MINOR=2  
ACTION=add  
SUBSYSTEM=block
```


Device Drivers

- La interface/software a bajo nivel de los dispositivos que caen en los siguientes grupos:
 - Character devices
 - Aquellos archivos en los cuales se puede leer y/o escribir (pantalla, teclado, puertos paralelos/seriales)
 - Block devices
 - Dispositivos que sólo puedes leer o escribir en múltiplos del tamaño de los bloques. Dispositivos en los cuales puedes montar un sistema de archivos. (/dev/sda)
 - Network interfaces

Block Devices

- Se encuentran en /dev y hacen referencia a varios dispositivos.
- Son identificados mediante 2 números: *major number* y *minor number*
- El *major number* indica que device driver se debe utilizar
- El *minor number* se refiere a una instancia del dispositivo, el significado varía.

Block Devices

- Por lo general el *major number* 8 se refiere a dispositivos de disco
- No son legibles, necesitas montarlos primero. Para eso se necesita un *device driver* que sepa como trabajar con el filesystem usado.
- Sólomente root puede montar dispositivos
- Los dispositivos tienen permisos distintos.

top
+frenzy
a collaboration of imaginations
throwing
and you only thought that you knew
everything I have right at you
looking
the forgotten moment
following
you must decide what you believe
making the choice
is all that matters

We're getting close

- Comenzemos a juntar las piezas





```
stealth@noname:~  
File Edit View Terminal Tabs Help  
[stealth@noname ~]$ cc youdev.c -lex2fs  
[stealth@noname ~]$ ls -la /dev/random  
crw-rw-rw- 1 root root 1, 8 2009-04-09 22:10 /dev/random  
[stealth@noname ~]$ ls -la /dev/sda7  
brw-rw---- 1 root disk 8, 7 2009-04-09 22:10 /dev/sda7  
[stealth@noname ~]$ id  
uid=500(stealth) gid=500(stealth) groups=500(stealth) context=unconfined_u:unconfined_r:unconfined_t:s0  
[stealth@noname ~]$ ./a.out 8 7  
My PID: 2954  
Hit @pid 603  
found 500 10349 100700  
Made boomsh suid-root. Clearing disk-cache now. This can take a while ...  
..  
done!  
bash-3.2# id  
uid=0(root) gid=0(root) groups=500(stealth) context=unconfined_u:unconfined_r:unconfined_t:s0  
bash-3.2# uname -a  
Linux noname 2.6.27.5-117.fc10.i686 #1 SMP Tue Nov 18 12:19:59 EST 2008 i686 athlon i386 GNU/Linux  
bash-3.2# grep lp /etc/shadow  
lp:!:14202:0:99999:7:::  
bash-3.2#
```

Screenshot by Sebsatian Krahmer <http://c-skills.blogspot.com/>

The Start Line - 16-04-2009

- *“The first problem appears since the origin of KOBJECT_UEVENT messages are not verified, so any user can spoof messages that udevd takes as granted from kernel. This allows some trickery to hijack any blockdevice through its minor and major numbers and give it permission 0666. The rest is code.”*

– From Sebastian Krahmer's Blog

udev trickery

- udev no verifica que los mensajes UEVENT en realidad vengan del kernel.
- Podemos enviarle un mensaje UEVENT y udev nos obedecerá

```
add @uritonto  
DEVPATH=/dev/random  
MAJOR=8  
MINOR=3  
ACTION=add  
SUBSYSTEM=block
```

PoC

- Construimos un PoC que se comuniqué con udev y agregue un device con el nombre, mayor y minor number que queramos.

```
$ ls -l /dev/sda3 /dev/random
```

```
crw-rw-rw- 1 root root 1, 8 Aug 11 18:22 /dev/random
```

```
brw-rw---- 1 root disk 8, 3 Aug 11 18:22 /dev/sda3
```

```
$ ./udev -p 3402 -d /dev/random -m 8 -n 3
```

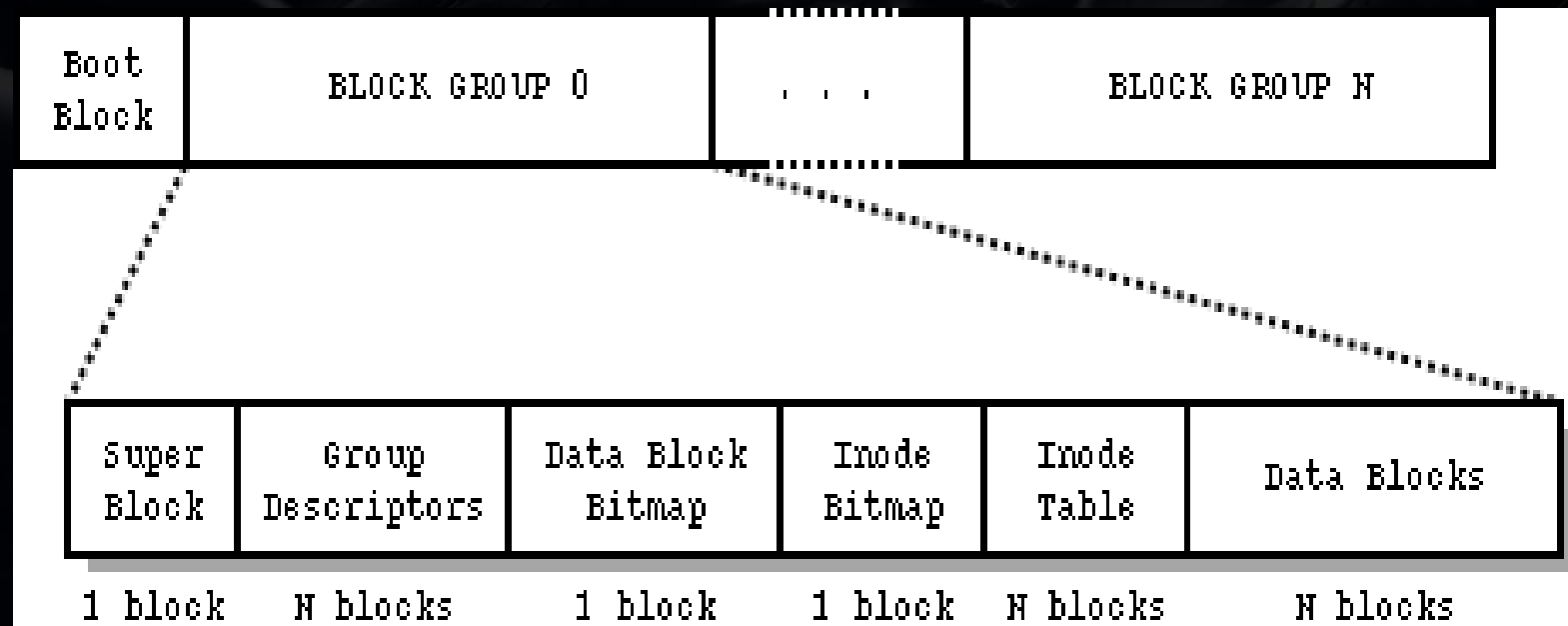
```
crw-rw-rw- 1 root root 8, 3 Aug 11 18:22 /dev/random
```

```
brw-rw---- 1 root disk 8, 3 Aug 11 18:25 /dev/sda3
```

Almost there...

- Ahora tenemos un device con permisos de lectura y escritura que apunta hacia el filesystem → Tenemos control entero sobre él
- ¿Cómo podemos modificarlo?
- No podemos montarlo
- No podemos leerlo
- No podemos programar un device driver

Ext2 filesystem structure



+frenzy
a collaboration of imaginations
everything is a false image
you must decide what you believe
making the choice
is all that matters

inode table

- La tabla de *inodes* contiene toda la información que el SO necesita saber acerca de cualquier archivo.
- Necesita ser accesada muy frecuentemente → disk cache
- Estas estructuras pueden ser modificadas únicamente por el kernel

/usr/include/linux/ext2_fs.h

```
struct ext2_inode {
    __u16  i_mode;          /* File type and access rights */
    __u16  i_uid;           /* Low 16 bits of Owner Uid */
    __u32  i_size;          /* Size in bytes */
    __u32  i_atime;         /* Access time */
    __u32  i_ctime;         /* Creation time */
    __u32  i_mtime;         /* Modification time */
    __u32  i_dtime;         /* Deletion Time */
    __u16  i_gid;           /* Low 16 bits of Group Id */
    __u16  i_links_count;   /* Links count */
    __u32  i_blocks;        /* Blocks count */
    __u32  i_flags;         /* File flags */
    ...
    __u32  i_block[EXT2_N_BLOCKS]; /* Pointers to blocks */
};
```

top
everything is a false image
you must decide what you believe
making the choice
is all that matters
do you know who you are?
throwing
a collaboration of imaginations
and you only thought that you knew
looking at the forgotten moment
following
V1

¿Cómo modificar las estructuras?

- ¿FileSystem Debugger?
- /sbin/debugfs
 - Debugger del sistema de archivos ext2
- ext2fs.h
 - Librería para C que permite interactuar con el sistema de archivos a bajo nivel

Ganando el privilegio

- Creando una puerta de acceso:

```
int main(){ setreuid(0);  
system("/bin/sh");}
```

- Modificamos sus permisos y le agregamos bit de SUID. (04755)
- Le donamos el archivo a root (id=0)

No hay cambio

- Modificamos la estructura pero el cache no se ha actualizado.
- Sólo root puede limpiar el cache.

- Forzamos un cache flush llenando el cache.

```
$ find / > /dev/null 2>/dev/null
```

```
$ find / -type f -exec cat {} \; > /dev/null
```


DEMO

- Código del exploit en:

<http://trew.icenetx.net/code/>



Got root?

- No puedes explotar algo sin saber precisamente como funciona.
- Mientras más conocimientos tienes sobre sistemas operativos, más maneras encuentras de explotar algo

top
+frenzy
a collaboration of imaginations
throwing
and you only thought that you knew
everything I have right at you
looking
at the forgotten moment
do you know who you are?
following
you must decide what you believe
making the choice
is all that matters

Gracias ¿Preguntas?

Ruben Ventura Piña
trew.revolution[at]gmail.com
<http://trew.icenetx.net>

Referencias

- <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-1185>
- <http://c-skills.blogspot.com/2009/04/udev-trickery-cve-2009-1185-and-cve.html>
- <http://pwnie-awards.org/2009/awards.html>
- <http://www.gentoo.org/security/en/glsa/glsa-200904-18.xml>

top
+frenzy
a collaboration of imaginations
throwing
and you only thought that you knew
everything I have right at you
looking at the forgotten moment
do you know who you are?
following
you must decide what you believe
making the choice
is all that matters

Referencias

- <http://en.wikipedia.org/wiki/Udev>
- <http://www.linuxjournal.com/article/7356>
- <http://www.cyberciti.biz/tips/understanding-unixlinux>
- <http://www.linuxjournal.com/article/7356>
- <http://maven.smith.edu/~nhowe/Teaching/csc262/oldlabs/ext2.html>