02.Debugging kernel and modules

Unknown macro: 'html'

Excuse the ads! We need some help to keep our site up. Unknown macro: 'html'

List

- Debugging kernel and modules
- Debug Symbol Packages
  - Getting -dbgsym.ddeb packages
  - Manual install of debug packages
- Disable KASLR
- Debugging Kernel and Modules with VMware
  - Debugging Preferences to VMware - debugStub
  - Connecting Debugging to the Kernel
  - Kernel Address Display Restriction(KADR)
  - perf_event_paranoid
  - Get section address of module
  - Debugging Modules
  - Debugging Preferences to VMware - serial port
- References

Debugging kernel and modules

- **, VMware .**

Debug Symbol Packages

- **Debug Symbol .**

Getting -dbgsym.ddeb packages

- "/etc/apt/sources.list.d/ddebs.list" .

Create an /etc/apt/sources.list.d/ddebs.list

```
echo "deb http://ddebs.ubuntu.com $(lsb_release -cs) main restricted universe multiverse
deb http://ddebs.ubuntu.com $(lsb_release -cs)-updates main restricted universe multiverse
deb http://ddebs.ubuntu.com $(lsb_release -cs)-proposed main restricted universe multiverse" | \
sudo tee -a /etc/apt/sources.list.d/ddebs.list
```

- Ubuntu Debug Symbol

Ubuntu 18.04 LTS

```
sudo apt install ubuntu-dbgsym-keyring
```

Manual install of debug packages

- Debug packages .

Install Debug Symball

```
sudo apt-get update
sudo apt-get install linux-image-$(uname -r)-dbgsym
```

- Debug Symbol .

Path to Debug symbol

```
/usr/lib/debug/boot/vmlinux-$(uname -r)
```

Disable KASLR

- **"/etc/default/grub" "GRUB_CMDLINE_LINUX_DEFAULT" "nokaslr" KASLR .**

/etc/default/grub

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet nokaslr"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

- .

```
sudo update-grub
```

Debugging Kernel and Modules with VMware

Debugging Preferences to VMware - debugStub

- **VMware VMware *.vmx .**
  - (32bit, 64bit) .

x86

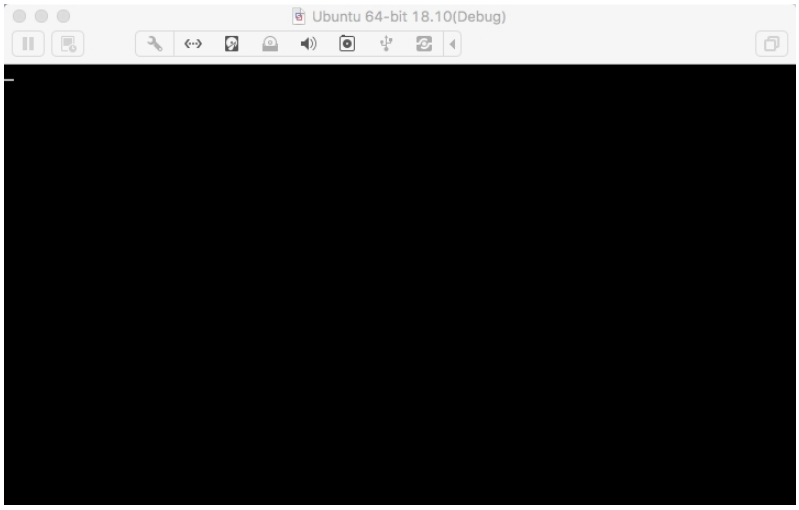
```
debugStub.listen.guest32 = "TRUE"
debugStub.listen.guest32.remote = "TRUE"
debugStub.hideBreakpoints = "FALSE"
monitor.debugOnStartGuest32 = "TRUE"
```

x64

```
debugStub.listen.guest64 = "TRUE"
debugStub.listen.guest64.remote = "TRUE"
debugStub.hideBreakpoints = "FALSE"
monitor.debugOnStartGuest64 = "TRUE"
```

Connecting Debugging to the Kernel

- VMware .



- **VM Root GDB VM .**
  - gdb architecture .
  - VMware , gdb "continue" .
- **gdb Port .**
  - 32bit : 8832
  - 64bit : 8864

Connect GDB

```
root@ubuntu:/home/lazenca0x0# gdb -q /usr/lib/debug/boot/vmlinux-4.18.0-12-generic
Reading symbols from /usr/lib/debug/boot/vmlinux-4.18.0-12-generic...done.
(gdb) set disassembly-flavor intel
(gdb) set architecture i386:x86-64:intel
The target architecture is assumed to be i386:x86-64:intel
(gdb) target remote 192.168.2.44:8864
Remote debugging using 192.168.2.44:8864
0x0000000001000200 in ?? ()
(gdb) c
Continuing.
```

Kernel Address Display Restriction(KADR)

- **"Kernel Address Display Restriction(KADR)" .**
  - Exploit User Space Exploit , , .

- **KADR , .**
- **.**
  - /boot/vmlinuz*, /boot/System.map*, /sys/kernel/debug/, /proc/slabinfo
- Ubuntu 11.04 "/proc/sys/kernel/kptr_rr_rrestrict" "1" .
  - "0" .

- **"/proc/kallsyms" "0000000000000000" .**
  - Root .

Regular local users

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ cat /proc/kallsyms |grep escalation
0000000000000000 t chardev_release        [escalation]
0000000000000000 t chardev_open        [escalation]
0000000000000000 t chardev_write        [escalation]
0000000000000000 t chardev_read        [escalation]
0000000000000000 t chardev_ioctl        [escalation]
0000000000000000 b info        [escalation]
0000000000000000 t chardev_init        [escalation]
0000000000000000 b chardev_cdev        [escalation]
0000000000000000 b chardev_major        [escalation]
0000000000000000 b __key.28909        [escalation]
0000000000000000 b chardev_class        [escalation]
0000000000000000 t chardev_exit        [escalation]
0000000000000000 d __this_module        [escalation]
0000000000000000 t cleanup_module        [escalation]
0000000000000000 t init_module        [escalation]
0000000000000000 d s_chardev_fops        [escalation]
lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

Root user

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo cat /proc/kallsyms |grep escalation
ffffffffc0301000 t chardev_release        [escalation]
ffffffffc0301019 t chardev_open        [escalation]
ffffffffc0301032 t chardev_write        [escalation]
ffffffffc0301051 t chardev_read        [escalation]
ffffffffc0301070 t chardev_ioctl        [escalation]
ffffffffc0303440 b info        [escalation]
ffffffffc0301154 t chardev_init        [escalation]
ffffffffc03034e0 b chardev_cdev        [escalation]
ffffffffc0303548 b chardev_major        [escalation]
ffffffffc0303440 b __key.28909        [escalation]
ffffffffc03034c8 b chardev_class        [escalation]
ffffffffc03012ac t chardev_exit        [escalation]
ffffffffc0303100 d __this_module        [escalation]
ffffffffc03012ac t cleanup_module        [escalation]
ffffffffc0301154 t init_module        [escalation]
ffffffffc0303000 d s_chardev_fops        [escalation]
lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

- **"kptr_restrict" "0" .**

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo sysctl -w kernel.kptr_restrict=0
kernel.kptr_restrict = 0
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ cat /proc/kallsyms |grep escalation
0000000000000000 t chardev_release        [escalation]
0000000000000000 t chardev_open        [escalation]
0000000000000000 t chardev_write        [escalation]
0000000000000000 t chardev_read        [escalation]
0000000000000000 t chardev_ioctl        [escalation]
0000000000000000 b info        [escalation]
0000000000000000 t chardev_init        [escalation]
0000000000000000 b chardev_cdev        [escalation]
0000000000000000 b chardev_major        [escalation]
0000000000000000 b __key.28909        [escalation]
0000000000000000 b chardev_class        [escalation]
0000000000000000 t chardev_exit        [escalation]
0000000000000000 d __this_module        [escalation]
0000000000000000 t cleanup_module        [escalation]
0000000000000000 t init_module        [escalation]
0000000000000000 d s_chardev_fops        [escalation]
lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

perf_event_paranoid

- **"kptr_restrict" "0" "perf_event_paranoid" .**
- **"perf_event_paranoid" Performance counters( ) .**

Option

Value	Description
2	(Linux 4.6)
1	(Linux 4.6)
0	CPU
-1	

- **"perf_event_paranoid" 1 0, -1 , .**

sudo sysctl -w kernel.perf_event_paranoid=0

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo sysctl kernel.perf_event_paranoid
kernel.perf_event_paranoid = 3
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo sysctl -w kernel.perf_event_paranoid=0
kernel.perf_event_paranoid = 0
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ cat /proc/kallsyms |grep escalation
ffffffffc0a5e000 t chardev_release        [escalation]
ffffffffc0a5e019 t chardev_open        [escalation]
ffffffffc0a5e032 t chardev_write        [escalation]
ffffffffc0a5e051 t chardev_read        [escalation]
ffffffffc0a5e070 t chardev_ioctl        [escalation]
ffffffffc0a60440 b info        [escalation]
ffffffffc0a5e154 t chardev_init        [escalation]
ffffffffc0a604e0 b chardev_cdev        [escalation]
ffffffffc0a60548 b chardev_major        [escalation]
ffffffffc0a60440 b __key.28909        [escalation]
ffffffffc0a604c8 b chardev_class        [escalation]
ffffffffc0a5e2ac t chardev_exit        [escalation]
ffffffffc0a60100 d __this_module        [escalation]
ffffffffc0a5e2ac t cleanup_module        [escalation]
ffffffffc0a5e154 t init_module        [escalation]
ffffffffc0a60000 d s_chardev_fops        [escalation]
lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

- https://wiki.ubuntu.com/Security/Features

Get section address of module

- **04.Creating a kernel module to privilege escalation** **"escalation.ko" .**
- .

Register a module

```
lazenca0x0@ubuntu:~$ cd Kernel/Module/escalation/
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo insmod escalation.ko
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ lsmod |grep escalation
escalation             16384  0
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ ls -al /dev/chardev0
crw-r--r-- 1 root root 240, 0 Dec 14 01:10 /dev/chardev0
lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

- **"/proc/kallsyms" .**
  - "/sys/module/" ".text", ".bss", ".data" , .
    - "/sys/module/( )/sections/.text.unlikely"
    - "/sys/module/( )/sections/.bss"
    - "/sys/module/( )/sections/.data"

Get section address of module

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo cat /proc/kallsyms |grep escalation
ffffffffc0301000 t chardev_release        [escalation]
ffffffffc0301019 t chardev_open        [escalation]
ffffffffc0301032 t chardev_write        [escalation]
ffffffffc0301051 t chardev_read        [escalation]
ffffffffc0301070 t chardev_ioctl        [escalation]
ffffffffc0303440 b info        [escalation]
ffffffffc0301154 t chardev_init        [escalation]
ffffffffc03034e0 b chardev_cdev        [escalation]
ffffffffc0303548 b chardev_major        [escalation]
ffffffffc0303440 b __key.28909        [escalation]
ffffffffc03034c8 b chardev_class        [escalation]
ffffffffc03012ac t chardev_exit        [escalation]
ffffffffc0303100 d __this_module        [escalation]
ffffffffc03012ac t cleanup_module        [escalation]
ffffffffc0301154 t init_module        [escalation]
ffffffffc0303000 d s_chardev_fops        [escalation]
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo cat /sys/module/escalation/sections/.text.unlikely
0xffffffffc0301000
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo cat /sys/module/escalation/sections/.bss
0xffffffffc0303440
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ sudo cat /sys/module/escalation/sections/.data
0xffffffffc0303000
lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

- **"chardev_ioctl" , .**

Disassembly the chardev_ioctl function

```
(gdb) c
Continuing.
^C
Program received signal SIGINT, Interrupt.
0xffffffff819f1ac6 in native_safe_halt () at /build/linux-ChQIyb/linux-4.18.0/arch/x86/include/asm/irqflags.h:57
57        /build/linux-ChQIyb/linux-4.18.0/arch/x86/include/asm/irqflags.h: No such file or directory.
(gdb) x/10i 0xffffffffc0301070
   0xffffffffc0301070:        nop    DWORD PTR [rax+rax*1+0x0]
   0xffffffffc0301075:        push   rbp
   0xffffffffc0301076:        mov    rdi,0xffffffffc03020e8
   0xffffffffc030107d:        mov    rbp,rsp
   0xffffffffc0301080:        push   r12
   0xffffffffc0301082:        mov    r12,rdx
   0xffffffffc0301085:        push   rbx
   0xffffffffc0301086:        mov    ebx,esi
   0xffffffffc0301088:        call   0xffffffff810f4e33 <printk>
   0xffffffffc030108d:        cmp    ebx,0x40884702
(gdb)
```

- **objdump "chardev_ioctl" .**

objdump -d escalation.ko

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ objdump -M intel -d escalation.ko

escalation.ko:     file format elf64-x86-64

Disassembly of section .text.unlikely:

0000000000000000 <chardev_release>:
   0:        e8 00 00 00 00               callq  5 <chardev_release+0x5>
...

0000000000000019 <chardev_open>:
  19:        e8 00 00 00 00               callq  1e <chardev_open+0x5>
...
0000000000000032 <chardev_write>:
  32:        e8 00 00 00 00               callq  37 <chardev_write+0x5>
...

0000000000000051 <chardev_read>:
  51:        e8 00 00 00 00               callq  56 <chardev_read+0x5>
...

0000000000000070 <chardev_ioctl>:
  70:        e8 00 00 00 00               call   75 <chardev_ioctl+0x5>
  75:        55                           push   rbp
  76:        48 c7 c7 00 00 00 00         mov    rdi,0x0
  7d:        48 89 e5                     mov    rbp,rsp
  80:        41 54                        push   r12
  82:        49 89 d4                     mov    r12,rdx
  85:        53                           push   rbx
  86:        89 f3                        mov    ebx,esi
  88:        e8 00 00 00 00               call   8d <chardev_ioctl+0x1d>
  8d:        81 fb 02 47 88 40            cmp    ebx,0x40884702
  93:        74 36                        je     cb <chardev_ioctl+0x5b>
  95:        81 fb 03 47 88 80            cmp    ebx,0x80884703
  9b:        74 71                        je     10e <chardev_ioctl+0x9e>
  9d:        81 fb 00 47 00 00            cmp    ebx,0x4700
  a3:        0f 85 8e 00 00 00            jne    137 <chardev_ioctl+0xc7>
  a9:        48 c7 c7 00 00 00 00         mov    rdi,0x0
  b0:        e8 00 00 00 00               call   b5 <chardev_ioctl+0x45>
  b5:        31 ff                        xor    edi,edi
  b7:        e8 00 00 00 00               call   bc <chardev_ioctl+0x4c>
  bc:        48 89 c7                     mov    rdi,rax
```

```
  bf:       e8 00 00 00 00          call   c4 <chardev_ioctl+0x54>
  c4:       31 d2                   xor    edx,edx
  c6:       e9 81 00 00 00          jmp    14c <chardev_ioctl+0xdc>
  cb:       48 c7 c7 00 00 00 00    mov    rdi,0x0
  d2:       e8 00 00 00 00          call   d7 <chardev_ioctl+0x67>
  d7:       ba 88 00 00 00          mov    edx,0x88
  dc:       4c 89 e6                mov    rsi,r12
  df:       48 c7 c7 00 00 00 00    mov    rdi,0x0
  e6:       e8 00 00 00 00          call   eb <chardev_ioctl+0x7b>
  eb:       48 85 c0                test   rax,rax
  ee:       75 55                   jne    145 <chardev_ioctl+0xd5>
  f0:       48 8b 35 00 00 00 00    mov    rsi,QWORD PTR [rip+0x0]        # f7 <chardev_ioctl+0x87>
  f7:       48 c7 c2 00 00 00 00    mov    rdx,0x0
  fe:       48 c7 c7 00 00 00 00    mov    rdi,0x0
 105:       e8 00 00 00 00          call   10a <chardev_ioctl+0x9a>
 10a:       31 d2                   xor    edx,edx
 10c:       eb 3e                   jmp    14c <chardev_ioctl+0xdc>
 10e:       48 c7 c7 00 00 00 00    mov    rdi,0x0
 115:       e8 00 00 00 00          call   11a <chardev_ioctl+0xaa>
 11a:       ba 88 00 00 00          mov    edx,0x88
 11f:       48 c7 c6 00 00 00 00    mov    rsi,0x0
 126:       4c 89 e7                mov    rdi,r12
 129:       e8 00 00 00 00          call   12e <chardev_ioctl+0xbe>
 12e:       31 d2                   xor    edx,edx
 130:       48 85 c0                test   rax,rax
 133:       75 10                   jne    145 <chardev_ioctl+0xd5>
 135:       eb 15                   jmp    14c <chardev_ioctl+0xdc>
 137:       89 de                   mov    esi,ebx
 139:       48 c7 c7 00 00 00 00    mov    rdi,0x0
 140:       e8 00 00 00 00          call   145 <chardev_ioctl+0xd5>
 145:       48 c7 c2 f2 ff ff ff    mov    rdx,0xfffffffffffffff2
 14c:       5b                      pop    rbx
 14d:       48 89 d0                mov    rax,rdx
 150:       41 5c                   pop    r12
 152:       5d                      pop    rbp
 153:       c3                      ret

0000000000000154 <init_module>:
...

00000000000002ac <cleanup_module>:
...

lazenca0x0@ubuntu:~/Kernel/Module/escalation$
```

Debugging Modules

- **"chardev_ioctl" Break point .**

Set Break point

```
(gdb) b *0xffffffffc0301075
Breakpoint 1 at 0xffffffffc0301075
(gdb) c
Continuing.
```

- **"Exploit" .**

Run Exploit

```
lazenca0x0@ubuntu:~/Kernel/Module/escalation$ ./Exploit
```

- **Breakpoint .**
  - .

Breakpoint 1, 0xffffffffc0301075 in ??

```
Breakpoint 1, 0xffffffffc0301075 in ?? ()
(gdb) i r rip
rip            0xffffffffc0301075  0xffffffffc0301075
(gdb) x/22i $rip
=> 0xffffffffc0301075:        push   rbp
   0xffffffffc0301076:        mov    rdi,0xffffffffc03020e8
   0xffffffffc030107d:        mov    rbp,rsp
   0xffffffffc0301080:        push   r12
   0xffffffffc0301082:        mov    r12,rdx
   0xffffffffc0301085:        push   rbx
   0xffffffffc0301086:        mov    ebx,esi
   0xffffffffc0301088:        call   0xffffffff810f4e33 <printk>
   0xffffffffc030108d:        cmp    ebx,0x40884702
   0xffffffffc0301093:        je     0xffffffffc03010cb
   0xffffffffc0301095:        cmp    ebx,0x80884703
   0xffffffffc030109b:        je     0xffffffffc030110e
   0xffffffffc030109d:        cmp    ebx,0x4700
   0xffffffffc03010a3:        jne    0xffffffffc0301137
   0xffffffffc03010a9:        mov    rdi,0xffffffffc03021e1
   0xffffffffc03010b0:        call   0xffffffff810f4e33 <printk>
   0xffffffffc03010b5:        xor    edi,edi
   0xffffffffc03010b7:        call   0xffffffff810b5a60 <prepare_kernel_cred>
   0xffffffffc03010bc:        mov    rdi,rax
   0xffffffffc03010bf:        call   0xffffffff810b56b0 <commit_creds>
   0xffffffffc03010c4:        xor    edx,edx
   0xffffffffc03010c6:        jmp    0xffffffffc030114c
(gdb)
```

- .
  - "cmp ebx,0x40884702" 0xffffffffc030108d Break point .
  - EBX 0x80884703 .
  - EBX 0x4700 , 0xffffffffc030109d .

Second brake point

```
(gdb) b *0xffffffffc030108d
Breakpoint 2 at 0xffffffffc030108d
(gdb) c
Continuing.

Breakpoint 2, 0xffffffffc030108d in ?? ()
(gdb) x/i $rip
=> 0xffffffffc030108d:        cmp    ebx,0x40884702
(gdb) i r ebx
ebx            0x4700              18176
(gdb) x/10i $rip
=> 0xffffffffc030108d:        cmp    ebx,0x40884702
   0xffffffffc0301093:        je     0xffffffffc03010cb
   0xffffffffc0301095:        cmp    ebx,0x80884703
   0xffffffffc030109b:        je     0xffffffffc030110e
   0xffffffffc030109d:        cmp    ebx,0x4700
   0xffffffffc03010a3:        jne    0xffffffffc0301137
   0xffffffffc03010a9:        mov    rdi,0xffffffffc03021e1
   0xffffffffc03010b0:        call   0xffffffff810f4e33 <printk>
   0xffffffffc03010b5:        xor    edi,edi
   0xffffffffc03010b7:        call   0xffffffff810b5a60 <prepare_kernel_cred>
(gdb)
```

- **"cmp ebx,0x4700" 0xffffffffc030109d Break point .**
  - EBX 0x4700 JNE RDI .
  - RDI 0xffffffffc03021e1 , "GIVE_ME_ROOT\n" .
  - , printk .

Third brake point

```
(gdb) b *0xffffffffc030109d
Breakpoint 3 at 0xffffffffc030109d
(gdb) c
Continuing.

Breakpoint 3, 0xffffffffc030109d in ?? ()
(gdb) x/i $rip
=> 0xffffffffc030109d:        cmp    ebx,0x4700
(gdb) i r ebx
ebx            0x4700              18176
(gdb) x/10i $rip
=> 0xffffffffc030109d:        cmp    ebx,0x4700
   0xffffffffc03010a3:        jne    0xffffffffc0301137
   0xffffffffc03010a9:        mov    rdi,0xffffffffc03021e1
   0xffffffffc03010b0:        call   0xffffffff810f4e33 <printk>
   0xffffffffc03010b5:        xor    edi,edi
   0xffffffffc03010b7:        call   0xffffffff810b5a60 <prepare_kernel_cred>
   0xffffffffc03010bc:        mov    rdi,rax
   0xffffffffc03010bf:        call   0xffffffff810b56b0 <commit_creds>
   0xffffffffc03010c4:        xor    edx,edx
   0xffffffffc03010c6:        jmp    0xffffffffc030114c
(gdb) si
0xffffffffc03010a3 in ?? ()
(gdb) si
0xffffffffc03010a9 in ?? ()
(gdb) x/i $rip
=> 0xffffffffc03010a9:        mov    rdi,0xffffffffc03021e1
(gdb) x/s 0xffffffffc03021e1
0xffffffffc03021e1:        "GIVE_ME_ROOT\n"
(gdb) si
0xffffffffc03010b0 in ?? ()
(gdb) si
printk (fmt=0xffffffffc03021e1 "GIVE_ME_ROOT\n") at /build/linux-ChQIyb/linux-4.18.0/kernel/printk/printk.c:1985
1985        in /build/linux-ChQIyb/linux-4.18.0/kernel/printk/printk.c
(gdb)
```

- **prepare_kernel_cred() .**
  - prepare_kernel_cred() RDI (0xffffffffc03010bc) Break point .
  - RAX prepare_kernel_cred() .
- **RAX struct cred uid, gid, suid, sgid, 0 .**
  - , Root .

Structure Data

```
(gdb) b *0xffffffffc03010bc
Breakpoint 4 at 0xffffffffc03010bc
(gdb) c
Continuing.

Breakpoint 4, 0xffffffffc03010bc in ?? ()
(gdb) i r rax
rax            0xffff880101b3c900  -131937071806208
(gdb) p *(struct cred*)$rax
$3 = {usage = {counter = 1}, uid = {val = 0}, gid = {val = 0}, suid = {val = 0}, sgid = {
    val = 0}, euid = {val = 0}, egid = {val = 0}, fsuid = {val = 0}, fsgid = {val = 0},
  securebits = 0, cap_inheritable = {cap = {0, 0}}, cap_permitted = {cap = {4294967295, 63}},
  cap_effective = {cap = {4294967295, 63}}, cap_bset = {cap = {4294967295, 63}}, cap_ambient = {
    cap = {0, 0}}, jit_keyring = 1 '\001', session_keyring = 0x0 <irq_stack_union>,
  process_keyring = 0x0 <irq_stack_union>, thread_keyring = 0x0 <irq_stack_union>,
  request_key_auth = 0x0 <irq_stack_union>, security = 0xffff8801390132a8,
  user = 0xffffffff82453d40 <root_user>, user_ns = 0xffffffff82453de0 <init_user_ns>,
  group_info = 0xffffffff8245b1a8 <init_groups>, rcu = {next = 0x0 <irq_stack_union>,
    func = 0x0 <irq_stack_union>}}
(gdb) p (*(struct cred*)$rax).uid
$3 = {val = 0}
(gdb) p (*(struct cred*)$rax).gid
$4 = {val = 0}
(gdb) p (*(struct cred*)$rax).suid
$5 = {val = 0}
(gdb) p (*(struct cred*)$rax).sgid
$6 = {val = 0}
(gdb)
```

Debugging Preferences to VMware - serial port

- **VMware debugStub , Serial Port .**
- **VMware *.vmx .**

Server(Debug)

```
serial0.present = "TRUE"
serial0.fileType = "pipe"
serial0.fileName = "/private/tmp/com1"
serial0.tryNoRxLoss = "FALSE"
serial0.pipe.endPoint = "server"
```

Client(gdb)

```
serial0.present = "TRUE"
serial0.fileType = "pipe"
serial0.fileName = "/private/tmp/com1"
serial0.tryNoRxLoss = "FALSE"
serial0.pipe.endPoint = "client"
```

- **"/etc/default/grub" "GRUB_CMDLINE_LINUX_DEFAULT" "kgdbwait kgdboc/ttyS0,115200" .**

/etc/default/grub

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet nokaslr"
GRUB_CMDLINE_LINUX="find_preseed=/preseed.cfg auto noprompt priority=critical locale=en_US"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

- .

```
sudo update-grub
```

- **IP, Port Serial Port .**

```
(gdb) target remote /dev/ttyS0
```

References

- https://mirrors.edge.kernel.org/pub/linux/kernel/
- https://github.com/surajx/qemu-arm-linux/wiki/Compile-Linux,-BusyBox-for-ARM-and-load-it-using-QEMU
- http://www.berkes.ca/guides/linux_kernel.html
- https://stackoverflow.com/questions/28298220/kernel-module-no-debugging-symbols-found
- http://www.yonch.com/tech/84-debugging-the-linux-kernel-with-qemu-and-eclipse
- http://www.linux-magazine.com/Online/Features/Qemu-and-the-Kernel
- https://www.collabora.com/news-and-blog/blog/2017/03/13/kernel-debugging-with-qemu-overview-tools-available/
- https://vmsplice.net/~stefan/stefanha-kernel-recipes-2015.pdf
- https://events.static.linuxfound.org/sites/events/files/slides/Debugging%20the%20Linux%20Kernel%20with%20GDB.pdf
- https://superuser.com/questions/298826/how-do-i-uncompress-vmlinuz-to-vmlinux
- https://wiki.ubuntu.com/DebuggingProcedures#Kernel.2FDebugging.Kernel_Debugging_Scenarios
- https://sysprogs.com/VisualKernel/tutorials/setup/ubuntu/
- https://github.com/cirosantilli/linux-kernel-module-cheat#linux-kernel-gdb-scripts
- https://wiki.ubuntu.com/Debug%20Symbol%20Packages
- https://www.dcl.hpi.uni-potsdam.de/research/WRK/2011/01/running-wrk-on-mac-os-with-vmware-fusion/index.html
- https://linux.die.net/man/2/perf_event_open
- https://www.anquanke.com/post/id/85837

Unknown macro: 'html'