

INTRODUÇÃO À PESQUISA EM VULNERABILIDADES NO NÚCLEO DO LINUX

Anderson EDUARDO
anderson [at] allelesecurity [DOT] com [DOT] BR

Agenda



sobre

Parte 1

Arquitetura de computadores
MOTIVAÇÕES
Ambiente de pesquisa
ECOSISTEMA
Linux
SUBSISTEMAS
Ferramentas

Parte 2

VULNERABILIDADES
CONSIDERAÇÕES FINAIS

perguntas

SOBRe



Anderson EDUARDO
CEO @ allele security intelligence

TWITTER: andersoncod3
GITHUB: andersoncod3
BLOG: [HTTP://BLOG.ANDERSONCOD3.IO](http://blog.andersoncod3.io)

Anteriormente

ADMINISTRADOR DE SISTEMAS
ANALISTA DE SEGURANÇA DA INFORMAÇÃO
PESQUISADOR EM SEGURANÇA DA INFORMAÇÃO INDEPENDENTE

ESTUDA O KERNEL DO FREEBSD DESDE ~ 2010
ESTUDA O KERNEL DO LINUX DESDE 2015

PRINCIPAIS INTERESSES

VULNERABILITY RESEARCH, ROOTKITS, SOURCE CODE REVIEW,
OPERATING SYSTEMS, COMPUTER SCIENCE, MATHEMATICS, PHYSICS

AVISO Legal



ESTE É O RESULTADO DE UM TRABALHO PESSOAL E SEM REVISÕES PRÉVIAS, NASCIDO DO INTERESSE DO APRESENTADOR COMO UMA FORMA DE ESTUDAR ASSUNTOS DE SEU INTERESSE, SENDO ASSIM, ERROS PODERÃO SER COMETIDOS NESTA APRESENTAÇÃO E PEÇO DESCULPAS DESDE JÁ.

"A Arquitetura de computadores é o projeto conceitual e fundamental da estrutura operacional de um sistema computacional. Ela é o estudo dos requisitos necessários para que um computador funcione e de como organizar os diversos componentes para obter melhores desempenhos."

X86 X86-64/AMD64 ARM ARM64 SPARC Itanium MIPS POWERPC

MODOS DE OPERAÇÃO DE uma CPU X86



REAL mode

- 16 BITS

- sem anéis de proteção

- acesso TOTAL ao Hardware

PROTECTED mode

- RINGS

- segmentação

- Paginação

- 32 BITS - 4G

VIRTUAL 8086

- DOS

LONG mode

- Paginação

- SUPOrTE LIMITADO a segmentação

- AMD64 canonical address

- 64 BITS registradores

- SUB MODO DE COMPATIBILIDADE 32 BITS

SMM

- 16 BITS

- smram

- Gerenciador SMI PROVIDO pela BIOS/UEFI

- usado normalmente para gerenciamento de erros críticos

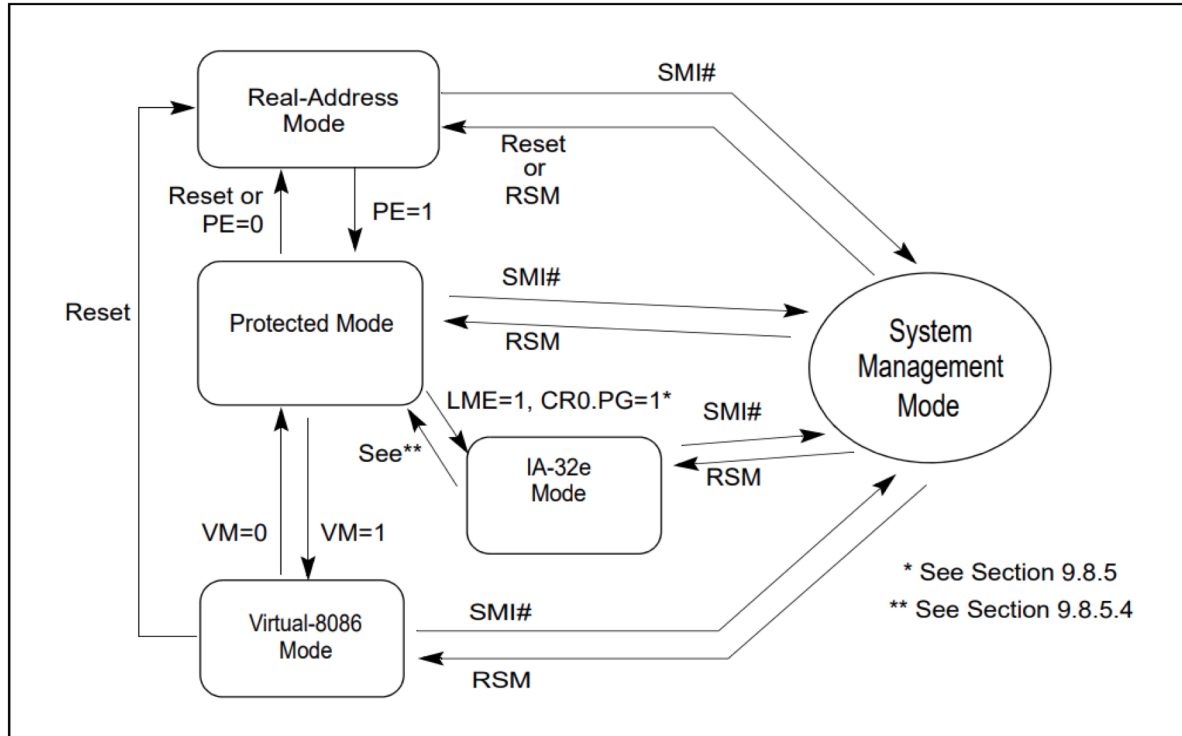
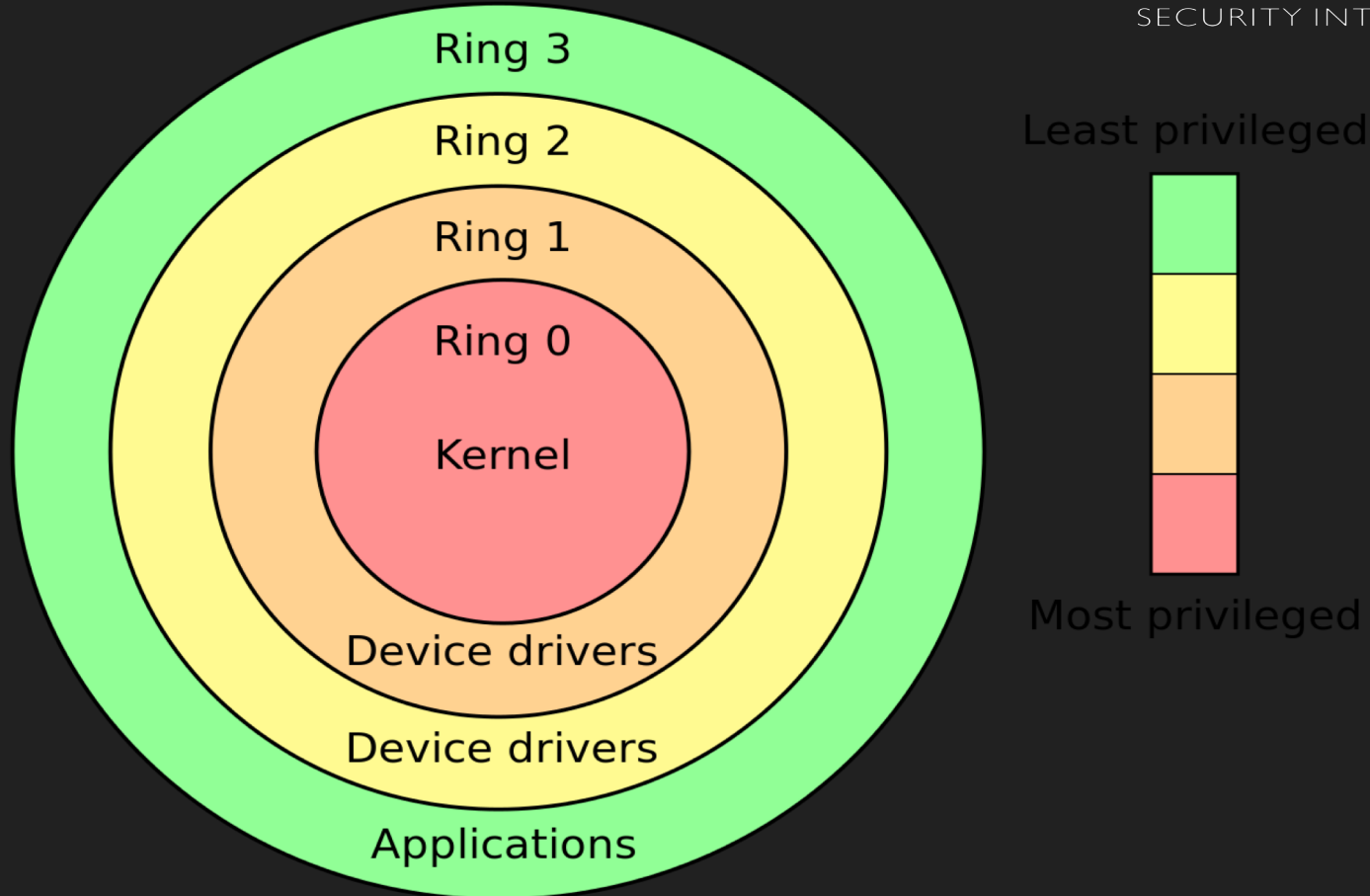
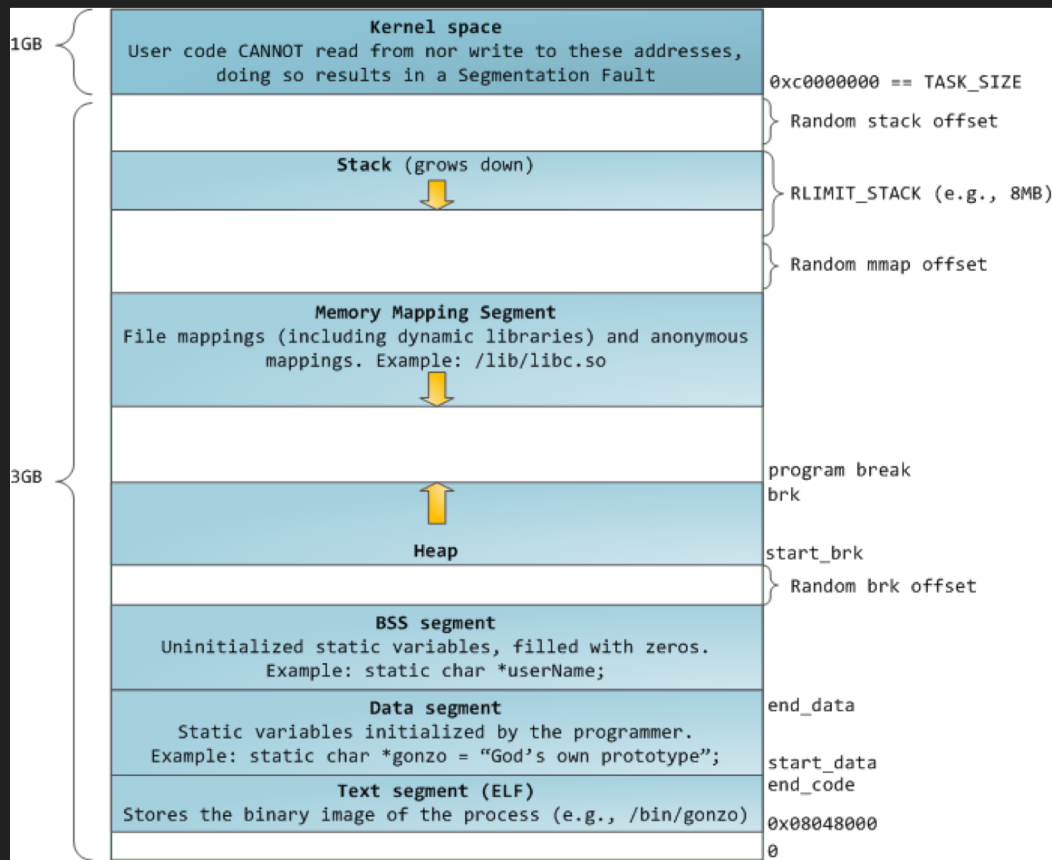


Figure 2-3. Transitions Among the Processor's Operating Modes

* See Section 9.8.5
** See Section 9.8.5.4



ESPAÇO DE endereçamento x86



MOTIVAÇÕES

Porque estudar o núcleo do Linux?

MOTIVAÇÕES

Porque estudar o núcleo do Linux?

conhecimento

MOTIVAÇÕES



Porque estudar o núcleo DO LINUX?

conhecimento

APrender como um núcleo funciona permite ao programador entender melhor a máquina.

MOTIVAÇÕES



Porque estudar o núcleo do Linux?

conhecimento

Oportunidade de emprego

MOTIVAÇÕES



Porque estudar o núcleo do Linux?

conhecimento

OPORTUNIDADE DE EMPREGO

Ter conhecimentos de como um núcleo funciona é um grande diferencial. Pode não haver muitas oportunidades mas as que existem podem ser bastante interessantes.

MOTIVAÇÕES



PORQUE ESTUDAR O NÚCLEO DO LINUX?

CONHECIMENTO

OPORTUNIDADE DE EMPREGO

RETORNO FINANCEIRO

MOTIVAÇÕES

Porque estudar o núcleo do LINUX?

conhecimento

OPORTUNIDADE DE EMPREGO

Retorno financeiro

Retorno financeiro através do mercado de compra e venda de vulnerabilidades.

PWN2OWN, \$15,000, UBUNTU

ZIMPERIUM, n-DAYS, Preço desconhecido

ZERODIUM, CENTOS UBUNTU TAILS, > \$30,000

INTERNET BUG BOUNTY, \$10,000, COMEX, TOWELROOT

MOTIVAÇÕES



PORQUE ESTUDAR O NÚCLEO DO LINUX?

CONHECIMENTO

OPORTUNIDADE DE EMPREGO

RETORNO FINANCEIRO

FURTIVIDADE

MOTIVAÇÕES

Porque estudar o núcleo do Linux?

conhecimento

oportunidade de emprego

retorno financeiro

furtividade

o núcleo é uma das principais partes de um sistema. se um atacante tem controle do núcleo, seu único limite será sua imaginação.

MOTIVAÇÕES

Porque estudar o núcleo do Linux?

conhecimento

Oportunidade de emprego

Retorno financeiro

Furtividade

Superfície de ataque/escalação de privilégios/sandbox

MOTIVAÇÕES

Porque estudar o núcleo do Linux?

conhecimento

Oportunidade de emprego

Retorno financeiro

Furtividade

Superfície de ataque/escalação de privilégios/sandbox

o kernel contém por padrão uma grande superfície de ataque.

AMBIENTE DE PESQUISA/ESTUDO



P: Qual ambiente mais adequado para se iniciar os estudos do núcleo do Linux?

AMBIENTE DE PESQUISA/ESTUDO



P: Qual ambiente mais adequado para se iniciar os estudos do núcleo do Linux?

R: Depende de cada pesquisador e suas motivações

AMBIENTE DE PESQUISA/ESTUDO



P: Qual ambiente mais adequado para se iniciar os estudos do núcleo do Linux?

R: Depende de cada pesquisador e suas motivações

Porém, existem algumas possibilidades:

AMBIENTE DE PESQUISA/ESTUDO



P: Qual ambiente mais adequado para se iniciar os estudos do núcleo do Linux?

R: Depende de cada pesquisador e suas motivações

Porém, existem algumas possibilidades:

MAQUINAS reais

AMBIENTE DE PESQUISA/ESTUDO



P: Qual ambiente mais adequado para se iniciar os estudos do núcleo do Linux?

R: Depende de cada pesquisador e suas motivações

Porém, existem algumas possibilidades:

Maquinas reais

Maquinas virtuais

AMBIENTE DE PESQUISA/ESTUDO



P: Qual ambiente mais adequado para se iniciar os estudos do núcleo do Linux?

R: Depende de cada pesquisador e suas motivações

Porém, existem algumas possibilidades:

Maquinas reais

Maquinas virtuais

Nuvem

AMBIENTE DE PESQUISA/ESTUDO



Maquinas reais

vantagem

segurança

desvantagem

gerenciamento

CUSTO

Hardware

MOBILIDADE

AMBIENTE DE PESQUISA/ESTUDO

Maquinas virtual

vantagem

segurança
CUSTO

Desvantagem

ESPAÇO em DISCO
MOBILIDADE

AMBIENTE DE PESQUISA/ESTUDO



nuvem

vantagem

MOBILIDADE
Hardware

desvantagem

segurança
CUSTO

ECOSISTEMA



LINUS TORVALDS

criador, mantenedor e PRINCIPAL desenvolvedor

GREG KROAH-HARTMAN

desenvolvedor e mantenedor de vários subsistemas
mantenedor do BRANCH -STABLE

DESENVOLVEDORES

EMPRESAS

RED HAT
GOOGLE
SUSE
CANONICAL
ORACLE

DISTRIBUIÇÕES

Empresas:

RED HAT

RED HAT ENTERPRISE LINUX
FEDORA
CENTOS

GOOGLE

ANDROID

SUSE

OPENSUSE

CANONICAL

UBUNTU

ORACLE

ORACLE LINUX

LINUX

kernel release 4.14.0

61,000 FILES

25,000,000 Lines

4,394 DEVELOPERS

≈530 COMPANIES kernel

releases 4.9.0 - 4.14.0

DECEMBER 2016 - NOVEMBER 2017

LINUX

10,000 Lines added

2,700 Lines removed

2,000 Lines modified kernel

releases 4.9.0 - 4.14.0

December 2016 - November 2017

Every day

como obter o núcleo do linux

GITHUB

`HTTP://GITHUB.COM/TORVALDS/LINUX`

`GIT CLONE HTTP://GITHUB.COM/TORVALDS/LINUX`

kernel.org

`HTTPS://GIT.KERNEL.ORG/`

`GIT CLONE`

`HTTPS://GIT.KERNEL.ORG/PUB/SCM/LINUX/KERNEL/GIT/TORVALDS/LINUX.GIT/`

GIT.CENTOS.ORG

`HTTPS://GIT.CENTOS.ORG/SUMMARY/RPMS!KERNEL.GIT`

`GIT CLONE HTTPS://GIT.CENTOS.ORG/GIT/RPMS/KERNEL.GIT && CD kernel && GIT
CHECKOUT C7 / GIT CHECKOUT [TAG]`

`GIT CLONE HTTPS://GIT.CENTOS.ORG/GIT/CENTOS-GIT-COMMON.GIT
GET SOURCESH`

continuous release - permite acesso previamente a pacotes que estarão disponíveis nas próximas versões

[AdditionalResources](#) » [Repositories](#) » [CR](#)

The Continuous Release (CR) Repository

Contents

1. Purpose
2. The Issue
3. A Resolution
4. Usage
5. Announcements

1. Purpose

The continuous release (CR) repository makes generally available packages that will appear in the next point release of CentOS, on a testing and **hotfix** basis until formally released.

2. The Issue

At a point release stage, new sources have been released by the upstream vendor, but binary updates are not formally issued as **complete** until the ISOs are ready (e.g., here, during the 5.8 release build process). This entails, in part building the updated RPMs with any needed patches; solving installer issues that crop up in **anaconda**; verification by the QA group of content; the delay while mirrors synchronize sufficiently to handle the update load; and finally formal release of a new point release.

For new major releases (i.e., the .0 point release), there is no prior point release in the major release series which may need security issues addressed, and so no need for streaming out 6.0 series updates before 6.0 itself was formally released; however for existing installs of a given major release, some administrators may desire, after review of announced updates from the upstream vendor, to update sooner than the complete CentOS point release rebuild process formerly permitted.

4. Usage

Usage of the **CR** repository is entirely optional to a system administrator, and is based on their express actions to opt-in.

Installation instructions for CentOS 5 and 6: You can install the CR repository by running **yum install centos-release-cr**. The package is included in the CentOS Extras repository, enabled by default.

Installation instructions for CentOS 7: The repository configuration file is included in the newest centos-release package. First update your system with **yum update** to get the new centos-release package, then run **yum-config-manager --enable cr** to enable the CR repository.

You can verify that your repository configuration is correct by running **yum repolist**. You should see a CR repository in the repository list. The CR repository will be empty and ignored until packages are placed in it at point release roll-over time. It will be emptied once the new point release is complete, but you can remain with the CR repository enabled if you so choose. This will allow you to be set up to receive CR updates for the next point release roll-over cycle.

ESTRUTURA DO NÚCLEO DO LINUX

LS -LA LINUX-3.10.0-693.21.1.el7-RESEARCH

drwxrwxr-x. 33 andersoncod3 andersoncod3	4096 Mar 2 15:19 arch
drwxrwxr-x. 3 andersoncod3 andersoncod3	4096 Mar 2 15:19 BLOCK
-rw-rw-r--. 1 andersoncod3 andersoncod3	140888 Mar 2 03:08 .CONFIG
drwxrwxr-x. 2 andersoncod3 andersoncod3	4096 Mar 2 15:19 CERTS
-rw-rw-r--. 1 andersoncod3 andersoncod3	18693 Mar 2 15:19 COPYING
-rw-rw-r--. 1 andersoncod3 andersoncod3	98556 Mar 2 15:19 CREDITS
drwxrwxr-x. 4 andersoncod3 andersoncod3	4096 Mar 2 15:19 CRYPTO
drwxrwxr-x. 124 andersoncod3 andersoncod3	12288 Mar 2 15:19 DOCUMENTATION
drwxrwxr-x. 136 andersoncod3 andersoncod3	4096 Mar 2 15:19 Drivers
drwxrwxr-x. 2 andersoncod3 andersoncod3	4096 Mar 2 15:19 Firmware
drwxrwxr-x. 73 andersoncod3 andersoncod3	4096 Mar 2 15:19 FS
drwxrwxr-x. 27 andersoncod3 andersoncod3	4096 Mar 2 15:19 INCLUDE
drwxrwxr-x. 2 andersoncod3 andersoncod3	4096 Mar 2 15:19 INIT
drwxrwxr-x. 2 andersoncod3 andersoncod3	4096 Mar 2 04:49 IPC
drwxrwxr-x. 17 andersoncod3 andersoncod3	4096 Mar 2 15:19 kernel
drwxrwxr-x. 13 andersoncod3 andersoncod3	12288 Mar 2 15:19 LIB
drwxrwxr-x. 5 andersoncod3 andersoncod3	4096 Mar 2 15:19 LICENSES
-rw-rw-r--. 1 andersoncod3 andersoncod3	445925 Mar 2 15:19 MAINTAINERS
-rw-rw-r--. 1 andersoncod3 andersoncod3	61760 Mar 2 15:19 Makefile
drwxrwxr-x. 3 andersoncod3 andersoncod3	4096 Mar 2 15:19 mm
drwxrwxr-x. 68 andersoncod3 andersoncod3	4096 Mar 2 15:19 net
-rw-rw-r--. 1 andersoncod3 andersoncod3	722 Mar 2 15:19 README
drwxrwxr-x. 28 andersoncod3 andersoncod3	4096 Mar 2 15:19 SAMPLES
drwxrwxr-x. 14 andersoncod3 andersoncod3	4096 Mar 2 15:19 SCRIPTS
drwxrwxr-x. 10 andersoncod3 andersoncod3	4096 Mar 2 15:19 SECURITY
drwxrwxr-x. 25 andersoncod3 andersoncod3	4096 Mar 2 15:19 SOUND
drwxrwxr-x. 31 andersoncod3 andersoncod3	4096 Mar 2 15:19 TOOLS
drwxrwxr-x. 2 andersoncod3 andersoncod3	4096 Mar 2 15:19 usr
drwxrwxr-x. 4 andersoncod3 andersoncod3	4096 Mar 2 15:19 VIRT

CONFIGURAÇÃO DO NÚCLEO DO LINUX



ARQUIVO .CONFIG

Make menuconfig

CONFIGURAÇÃO DO NÚCLEO

```
.config - Linux/x86 3.10.0-693.5.2.el7 Kernel Configuration

Linux/x86 3.10.0-693.5.2.el7 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

[*] 64-bit kernel
  General setup --->
  *- Provide system-wide ring of trusted keys
  [*] Provide system-wide ring of blacklisted keys
  [*] Enable loadable module support --->
  *- Enable the block layer --->
    Processor type and features --->
    Power management and ACPI options --->
    Bus options (PCI etc.) --->
    Executable file formats / Emulations --->
  <*> Volume Management Device Driver
  *- Networking support --->
    Device Drivers --->
    Firmware Drivers --->
    File systems --->
    Kernel hacking --->
    Security options --->
  *- Cryptographic API --->
  [*] Virtualization --->
    Library routines --->
  [*] Enables more stringent kabi checks in the macros

<Select> < Exit > < Help > < Save > < Load >
```

COMPILAÇÃO DO NÚCLEO DO LINUX

make

SUDO make modules_install install

SUBSISTEMAS



IPC (INTER PROCESS COMMUNICATION)

MM (MEMORY MANAGEMENT)

VFS (VIRTUAL FILE SYSTEMS)

NETWORKING

SECURITY

AUDITING

PROCESS

SCHEDULING

POWER MANAGEMENT

...

ALLELE
SECURITY INTELLIGENCE

[illegible]

© 2007, 2010 Constantine Shulyupin www.MakeLinux.net/kernel_map

Ferramentas

EDITOR DE TEXTO

vim

NAVEGADORES DE BASE DE CÓDIGO

CSCOPE

DIFFING

meld

Análise estática

smatch
coccinelle
sparse

DEBUGGING

GDB + vmware

Ferramentas

Dynamic Tracing

systemtap
kprobes

Fuzzers

syzkaller
trinity

Functions calling this function: __sys_recvmsg

File	Function	Line
0 net/compat.c	compat_sys_recvmsg	786 return __sys_recvmsg(fd, (struct mmsghdr __user *)mmsg, vlen,
1 net/compat.c	compat_sys_recvmsg	791 return __sys_recvmsg(fd, (struct mmsghdr __user *)mmsg, vlen,
2 net/compat.c	compat_sys_recvmsg	797 datagrams = __sys_recvmsg(fd, (struct mmsghdr __user *)mmsg, vlen,
3 net/socket.c	SYSCALL_DEFINES	2420 return __sys_recvmsg(fd, mmsg, vlen, flags, NULL);
4 net/socket.c	SYSCALL_DEFINES	2425 datagrams = __sys_recvmsg(fd, mmsg, vlen, flags, &timeout_sys);

Find this C symbol:

Find this global definition:

Find functions called by this function:

Find functions calling this function:

Find this text string:

Change this text string:

Find this egrep pattern:

Find this file:

Find files #including this file:

Find assignments to this symbol:

```
git clone git://repo.or.cz/smatch.git

cd smatch

make
```

Now let's create a small test.c file:

```
#include "check_debug.h"
int var;
void function(void)
{

    if (var < 0 || var >= 10)

    return;

    __smatch_implied(var);
}
```

The "check_debug.h" file can be included into any .c file. It is used to display internal Smatch information which helps with debugging. If you run `./smatch test.c`, then that prints the value of the "var" variable.

```
test.c:9 function() implied: var = '0-9'
```

PARTE 2

vulnerabilidades

CVE-2015-7613

"Race condition in the IPC object implementation in the Linux kernel through 4.2.3 allows local users to gain privileges by triggering an IPC_addid call that leads to uid and gid comparisons against uninitialized data, related to msg.c, shm.c, and util.c."

SVIPC(7)

Linux Programmer's Manual

SVIPC(7)

NAME

svipc - System V interprocess communication mechanisms

SYNOPSIS

```
#include <sys/msg.h>
#include <sys/sem.h>
#include <sys/shm.h>
```

DESCRIPTION

This manual page refers to the Linux implementation of the System V interprocess communication (IPC) mechanisms: message queues, semaphore sets and shared memory segments. In the following, the word resource means an instantiation of one among such mechanisms.

A vulnerabilidade afetava TODO o subsistema (3 mecanismos) mas quando ela foi corrigida inicialmente em upstream, somente em um mecanismo foi corrigido (semáforo). Após a correção, ainda era possível explorar a mesma vulnerabilidade utilizando os outros dois mecanismos do subsistema IPC (message, shared memory).

Initialize msg/shm IPC objects before doing ipc_addid()

[Browse files](#)

As reported by Dmitry Vyukov, we really shouldn't do `ipc_addid()` before having initialized the IPC object state. Yes, we initialize the IPC object in a locked state, but with all the lockless RCU lookup work, that IPC object lock no longer means that the state cannot be seen.

We already did this for the IPC semaphore code (see commit [e8577d1](#): "ipc/sem.c: fully initialize sem_array before making it visible") but we clearly forgot about msg and shm.

Reported-by: Dmitry Vyukov <dvyukov@google.com>

Cc: Manfred Spraul <manfred@colorfullife.com>

Cc: Davidlohr Bueso <dbueso@suse.de>

Cc: stable@vger.kernel.org

Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>

 master (#115)  v4.16 ... v4.3-rc4



torvalds committed on **Sep 30, 2015**

1 parent [3225031](#)

commit [b9a532277938798b53178d5a66af6e2915cb27cf](#)

ipc/sem.c: fully initialize sem_array before making it visible

[Browse files](#)

`ipc_addid()` makes a new ipc identifier visible to everyone. New objects start as locked, so that the caller can complete the initialization after the call. Within struct `sem_array`, at least `sma->sem_base` and `sma->sem_nsems` are accessed without any locks, therefore this approach doesn't work.

Thus: Move the `ipc_addid()` to the end of the initialization.

Signed-off-by: Manfred Spraul <manfred@colorfullife.com>

Reported-by: Rik van Riel <riel@redhat.com>

Acked-by: Rik van Riel <riel@redhat.com>

Acked-by: Davidlohr Bueso <dave@stgolabs.net>

Acked-by: Rafael Aquini <aquini@redhat.com>

Signed-off-by: Andrew Morton <akpm@linux-foundation.org>

Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>

 master (#4)  v4.16 ... v3.18



manfred-colorfu authored and torvalds committed on Dec 2, 2014 1 parent 92788ac commit e8577d1f0329d4842e8302e289fb2c22156abef4

LIÇÕES

Programadores experientes, também erram
(esquecem)

conhecimento do funcionamento interno do
subsistema ajuda bastante durante o processo de
procura, entendimento e exploração de
vulnerabilidades

CVE-2017-10661

"Race condition in fs/timerfd.c in the Linux kernel before 4.10.15 allows local users to gain privileges or cause a denial of service (list corruption or use-after-free) via simultaneous file-descriptor operations that leverage improper might_cancel queueing."

NAME

timerfd_create, timerfd_settime, timerfd_gettime - timers that notify via file descriptors

SYNOPSIS

```
#include <sys/timerfd.h>
```

```
int timerfd_create(int clockid, int flags);
```

```
int timerfd_settime(int fd, int flags,  
                    const struct itimerspec *new_value,  
                    struct itimerspec *old_value);
```

```
int timerfd_gettime(int fd, struct itimerspec *curr_value);
```

DESCRIPTION

These system calls create and operate on a timer that delivers timer expiration notifications via a file descriptor. They provide an alternative to the use of `setitimer(2)` or `timer_create(2)`, with the advantage that the file descriptor may be monitored by `select(2)`, `poll(2)`, and `epoll(7)`.

The use of these three system calls is analogous to the use of `timer_create(2)`, `timer_settime(2)`, and `timer_gettime(2)`. (There is no analog of `timer_getoverrun(2)`, since that functionality is provided by `read(2)`, as described below.)

vulnerabilidade publicamente conhecida que ainda afeta, pelo menos, a última versão do CentOS 7 (Linux-3.10.0-693.21.1.el7)

```
→ linux-3.10.0-693.21.1.el7 git:(187831e) ✗ uname -a
Linux research01 3.10.0-693.21.1.el7 #0 SMP Mon Mar 5 11:34:50 -03 2018 x86_64 x86_64 x86_64 GNU/Linux
→ linux-3.10.0-693.21.1.el7 git:(187831e) ✗ /tmp/poc > /dev/null
AC
→ linux-3.10.0-693.21.1.el7 git:(187831e) ✗ dmesg|grep -A3 WARNING|tail -f
--
[64423.042382] WARNING: CPU: 2 PID: 101718 at lib/list_debug.c:62 __list_del_entry+0x82/0xd0
[64423.042383] list_del corruption. next->prev should be ffff880079336ca8, but was dead000000000200
[64423.042384] Modules linked in: coretemp snd_seq_midi snd_seq_midi_event iosf_mbi crc32_pclmul ghash_clmulni_intel aesni_intel l
rw gf128mul glue_helper ablk_helper cryptd snd_ens1371 snd_rawmidi snd_ac97_codec ac97_bus snd_seq snd_seq_device snd_pcm snd_time
r nfit snd ppdev pcspkr joydev libnvdimm vmw_balloon parport_pc soundcore parport sg vmw_vmci shpchp i2c_piix4 ip_tables ext4 mbca
che jbd2 sr_mod cdrom ata_generic pata_acpi vmwgfx sd_mod crc_t10dif drm_kms_helper crct10dif_generic syscopyarea sysfillrect sysi
mgbt fb_sys_fops ttm drm ata_piix libata mptspi e1000 scsi_transport_spi mptscsih crct10dif_pclmul crct10dif_common mptbase crc32
c_intel i2c_core serio_raw
[64423.042417] CPU: 2 PID: 101718 Comm: poc Tainted: G          W          ----- 3.10.0-693.21.1.el7 #0
--
[64444.630195] WARNING: CPU: 1 PID: 26296 at lib/list_debug.c:91 __list_add_rcu+0x5b/0x80
[64444.630197] list_add_rcu corruption. next->prev should be prev (ffffffffff81a98d60), but was dead000000000200. (next=ffff88007933
6ca8).
[64444.630198] Modules linked in: coretemp snd_seq_midi snd_seq_midi_event iosf_mbi crc32_pclmul ghash_clmulni_intel aesni_intel l
rw gf128mul glue_helper ablk_helper cryptd snd_ens1371 snd_rawmidi snd_ac97_codec ac97_bus snd_seq snd_seq_device snd_pcm snd_time
r nfit snd ppdev pcspkr joydev libnvdimm vmw_balloon parport_pc soundcore parport sg vmw_vmci shpchp i2c_piix4 ip_tables ext4 mbca
che jbd2 sr_mod cdrom ata_generic pata_acpi vmwgfx sd_mod crc_t10dif drm_kms_helper crct10dif_generic syscopyarea sysfillrect sysi
mgbt fb_sys_fops ttm drm ata_piix libata mptspi e1000 scsi_transport_spi mptscsih crct10dif_pclmul crct10dif_common mptbase crc32
c_intel i2c_core serio_raw
[64444.630242] CPU: 1 PID: 26296 Comm: poc Tainted: G          W          ----- 3.10.0-693.21.1.el7 #0
→ linux-3.10.0-693.21.1.el7 git:(187831e) ✗ _
```

[about](#) [summary](#) [refs](#) [log](#) [tree](#) [commit](#) [diff](#) [stats](#)

author  Thomas Gleixner <tglx@linutronix.de> 2017-01-31 15:24:03 +0100
committer  Thomas Gleixner <tglx@linutronix.de> 2017-02-10 11:15:09 +0100
commit [1e38da300e1e395a15048b0af1e5305bd91402f6](#) (patch)
tree [9d5c20e4fb1aba203e59dbd6146eee723f258a3a](#)
parent [7551b02b94ad1daee3a79d667dc3c46d08328f87](#) (diff)
download [linux-1e38da300e1e395a15048b0af1e5305bd91402f6.tar.gz](#)

timerfd: Protect the might cancel mechanism proper

The handling of the `might_cancel` queueing is not properly protected, so parallel operations on the file descriptor can race with each other and lead to list corruptions or use after free.

Protect the context for these operations with a separate lock.

The wait queue lock cannot be reused for this because that would create a lock inversion scenario vs. the cancel lock. Replacing `might_cancel` with an atomic (`atomic_t` or atomic bit) does not help either because it still can race vs. the actual list operation.

Reported-by: Dmitry Vyukov <dvyukov@google.com>

Signed-off-by: Thomas Gleixner <tglx@linutronix.de>

Cc: "linux-fsdevel@vger.kernel.org"

Cc: syzkaller <syzkaller@googlegroups.com>

Cc: Al Viro <viro@zeniv.linux.org.uk>

Cc: linux-fsdevel@vger.kernel.org

Link: <http://lkml.kernel.org/r/alpine.DEB.2.20.1701311521430.3457@nanos>

Signed-off-by: Thomas Gleixner <tglx@linutronix.de>

First Last Prev Next This bug is not in your last search results.

[Format For Printing](#) - [XML](#) - [Clone This Bug](#) - [Last Comment](#)

Bug 1481136 -

Summary: CVE-2017-10661 kernel: Handling of might_cancel queueing is not properly pret...

Status: NEW

Aliases: CVE-2017-10661

Product: Security Response

Component: vulnerability ([Show other bugs](#))

Version: unspecified

Hardware: All Linux

Priority medium **Severity** medium

Target Milestone: ---

Target Release: ---

Assigned To: Red Hat Product Security

QA Contact:

Docs Contact:

URL:

Whiteboard: impact=moderate,public=20170210,repor...

Keywords: Security

Depends On: 1485404 1485405 1485406 1485407 1485408 1485409 1485410

Blocks: 1481154

[Show dependency tree / graph](#)

Reported: 2017-08-14 04:16 EDT [REDACTED]

Modified: 2018-02-07 18:36 EST ([History](#))

CC List: 37 users

See Also:

Fixed In Version:

Doc Type: Bug Fix

Doc Text: A race condition was found in the Linux kernel before version 4.11-rc1 in 'fs/timerfd.c' file which allows a local user to cause a kernel list corruption or use-after-free via simultaneous operations with a file descriptor which leverage improper 'might_cancel' queueing. An unprivileged local user could use this flaw to cause a denial of service of the system. Due to the nature of the flaw, privilege escalation cannot be fully ruled out, although we believe it is unlikely.

Clone Of:

Environment:

Last Closed:

Attachments [\(Terms of Use\)](#)

[Add an attachment](#) (proposed patch, testcase, etc.)

2017-08-14 04:16:55 EDT

Description

The handling of the might_cancel queueing is not properly protected, so parallel operations on the file descriptor can race with each other and lead to list corruptions or use after free.

References:

<https://marc.info/?l=linux-fsdevel&m=148587265720603&w=2>

<https://marc.info/?t=148587273100007&r=1&w=2>

<https://source.android.com/security/bulletin/2017-08-01#kernel-components>

Upstream patch:

<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit?id=1e38da300e1e395a15048b0af1e5305bd91402f6>

2017-08-25 11:20:24 EDT

Comment 4

Statement:

This issue does not affect Red Hat Enterprise Linux 5 as the code with the flaw is not present in the products listed.

This issue affects Red Hat Enterprise Linux 6, 7 and Red Hat Enterprise MRG 2. Future updates for the respective releases may address this issue.

Note

You need to [log in](#) before you can comment on or make changes to this bug.

LIÇÕES

BACKPORTING

VULNERABILIDADES CORRIGIDAS em UPSTream PODem
Levar TEMPO Para serem CORRIGIDAS nas DISTRIBUIÇÕES.

CVE-2016-7117

"Use-after-free vulnerability in the `_sys_recvmsg` function in `net/socket.c` in the Linux kernel before 4.5.2 allows remote attackers to execute arbitrary code via vectors involving a `recvmsg` system call that is mishandled during error processing."

NAME

recvmsg - receive multiple messages on a socket

SYNOPSIS

```
#define _GNU_SOURCE
#include <sys/socket.h>
```

```
int recvmsg(int sockfd, struct mmsghdr *msgvec, unsigned int vlen,
            unsigned int flags, struct timespec *timeout);
```

DESCRIPTION

The `recvmsg()` system call is an extension of `recvmsg(2)` that allows the caller to receive multiple messages from a socket using a single system call. (This has performance benefits for some applications.) A further extension over `recvmsg(2)` is support for a timeout on the receive operation.

The `sockfd` argument is the file descriptor of the socket to receive data from.

The `msgvec` argument is a pointer to an array of `mmsghdr` structures. The size of this array is specified in `vlen`.

The `mmsghdr` structure is defined in `<sys/socket.h>` as:

vulnerabilidade afetou uma parte principal do núcleo do linux, adicionada em 2009 e foi somente corrigida em 2016 (7 anos)

Functions calling this function: __sys_recvmsg

File	Function	Line
0 net/compat.c	compat_sys_recvmsg	786 return __sys_recvmsg(fd, (struct mmsghdr __user *)mmsg, vlen,
1 net/compat.c	compat_sys_recvmsg	791 return __sys_recvmsg(fd, (struct mmsghdr __user *)mmsg, vlen,
2 net/compat.c	compat_sys_recvmsg	797 datagrams = __sys_recvmsg(fd, (struct mmsghdr __user *)mmsg, vlen,
3 net/socket.c	SYSCALL_DEFINES	2420 return __sys_recvmsg(fd, mmsg, vlen, flags, NULL);
4 net/socket.c	SYSCALL_DEFINES	2425 datagrams = __sys_recvmsg(fd, mmsg, vlen, flags, &timeout_sys);

Find this C symbol:
Find this global definition:
Find functions called by this function:
Find functions calling this function:
Find this text string:
Change this text string:
Find this egrep pattern:
Find this file:
Find files #including this file:
Find assignments to this symbol:

net: Fix use after free in the recvmmsg exit path

[Browse files](#)

The syzkaller fuzzer hit the following use-after-free:

Call Trace:

```
[<ffffffff8175ea0e>] __asan_report_load8_noabort+0x3e/0x40 mm/kasan/report.c:295
[<ffffffff851cc31a>] __sys_recvmmsg+0x6fa/0x7f0 net/socket.c:2261
[<      inline      >] SYSC_recvmmsg net/socket.c:2281
[<ffffffff851cc57f>] SyS_recvmmsg+0x16f/0x180 net/socket.c:2270
[<ffffffff86332bb6>] entry_SYSCALL_64_fastpath+0x16/0x7a
arch/x86/entry/entry_64.S:185
```

And, as Dmitry rightly assessed, that is because we can drop the reference and then touch it when the underlying recvmmsg calls return some packets and then hit an error, which will make recvmmsg to set sock->sk->sk_err, oops, fix it.

Reported-and-Tested-by: Dmitry Vyukov <dvyukov@google.com>

Cc: Alexander Potapenko <glider@google.com>

Cc: Eric Dumazet <edumazet@google.com>

Cc: Kostya Serebryany <kcc@google.com>

Cc: Sasha Levin <sasha.levin@oracle.com>

Fixes: a2e2725 ("net: Introduce recvmmsg socket syscall")

<http://lkml.kernel.org/r/20160122211644.GC2470@redhat.com>

Signed-off-by: Arnaldo Carvalho de Melo <davem@redhat.com>

Signed-off-by: David S. Miller <davem@davemloft.net>

 master  v4.16 ... v4.6-rc1



Arnaldo Carvalho de Melo authored and davem330 committed on Mar 14, 2016 1 parent b6e4038 commit 34b88a68f26a75e4fded796f1a49c40f82234b7d

net: Introduce recvmmsg socket syscall

[Browse files](#)

Meaning receive multiple messages, reducing the number of syscalls and net stack entry/exit operations.

Next patches will introduce mechanisms where protocols that want to optimize this operation will provide an `unlocked_recvmmsg` operation.

This takes into account comments made by:

- . Paul Moore: `sock_recvmmsg` is called only for the first datagram, `sock_recvmmsg_nosec` is used for the rest.
- . Caitlin Bestler: `recvmmsg` now has a `struct timespec` timeout, that works in the same fashion as the `ppoll` one.

If the underlying protocol returns a datagram with `MSG_OOB` set, this will make `recvmmsg` return right away with as many datagrams (+ the `OOB` one) it has received so far.

- . Rémi Denis-Courmont & Steven Whitehouse: If we receive $N < vlen$ datagrams and then `recvmmsg` returns an error, `recvmmsg` will return the successfully received datagrams, store the error and return it in the next call.

This paves the way for a subsequent optimization, `sk_prot->unlocked_recvmmsg`, where we will be able to acquire the lock only at batch start and end, not at every underlying `recvmmsg` call.

Signed-off-by: Arnaldo Carvalho de Melo <acme@redhat.com>

Signed-off-by: David S. Miller <davem@davemloft.net>

 master  v4.16 ... v2.6.33-rc1



Arnaldo Carvalho de Melo authored and davem330 committed on Oct 13, 2009

1 parent [c05e85a](#)

commit [a2e2725541fad72416326798c2d7fa4dafb7d337](#)

vulnerabilidades - CVE-2016-7117

ALLELE

SECURITY INTELLIGENCE

Bug 1382268 -

Summary: CVE-2016-7117 kernel: Use-after-free in the recvmsg exit path

Status: CLOSED ERRATA

Aliases: CVE-2016-7117

Product: Security Response

Component: vulnerability ([Show other bugs](#))

Version: unspecified

Hardware: All Linux

Priority high **Severity** high

Target Milestone: ---

Target Release: ---

Assigned To: Red Hat Product Security

QA Contact:

Docs Contact:

URL:

Whiteboard: impact=important,public=20160314,repo...

Keywords: Security

Duplicates: [CVE-2017-8281](#) ([view as bug list](#))

Depends On: [1382269](#) [1390044](#) [1390046](#) [1390047](#) [1390048](#) [1390805](#) [1390806](#) [1390807](#) [1390808](#) [1399113](#) [1399114](#) [1399115](#) [1399116](#) [1399117](#) [1399118](#) [1399119](#)

Blocks: [1382270](#)

[Show dependency tree](#) / [graph](#)

Reported: 2016-10-06 03:54 EDT

Modified: 2017-09-07 04:53 EDT ([History](#))

CC List: 54 users

agordeev
alekcej
amaris
aquini
arm-mgr

See Also:

Fixed In Version:

Doc Type: If docs needed, set a value

Doc Text: A use-after-free vulnerability was found in the kernel's socket recvmsg subsystem. This may allow remote attackers to corrupt memory and may allow execution of arbitrary code. This corruption takes place during the error handling routines within __sys_recvmsg() function.

Clone Of:

Environment:

Last Closed: 2017-02-15 05:58:17 EST

um pouco mais que 3 meses para ser corrigida

errata-xmllrpc

2017-01-17 13:19:30 EST

[Comment 30](#)

This issue has been addressed in the following products:

Red Hat Enterprise Linux 7

Via RHSA-2017:0086 <https://rhn.redhat.com/errata/RHSA-2017-0086.html>

LIÇÕES

VALIDADE DA LEI DE LINUS (LINUS'S LAW)

"DADOS OLHOS SUFICIENTES, TODOS OS ERROS SÃO

ÓBVIOS

(GIVEN ENOUGH EYEBALLS, ALL BUGS ARE SHALLOW)"

PROFUNDIDADE DA BASE DE CÓDIGO

ESSA VULNERABILIDADE AFETOU UMA PARTE PRINCIPAL
DO NÚCLEO DO LINUX, A CHAMADA DE SISTEMA `recvmsg()`

considerações finais



MÓDULOS DE Terceiros

capAbILITIES

carregamento automático de módulos

Namespaces

CHAnGELoG / NOVOS recursos / Beta

LISTA DE DISCUSSões

OBRIGADO!

DÚVIDAS?