LINUX SECURITY SUMMIT

CVEhound

Denis Efremov / Oracle

@efrmv

THE LINUX FOUNDATION

What?

- CVEhound tool to detect absence of patches in the kernel sources
  - Project started in December 2020
  - GPLv3 for python code, GPLv2 for CVE detection rules
- Doesn’t look at kernel version
- Doesn’t require .git
- Doesn’t need build information

What?

- CVEhound tool to detect absence of patches in the kernel sources
  - Project started in December 2020
  - GPLv3 for python code, GPLv2 for CVE detection rules
- Doesn’t look at kernel version
- Doesn’t require .git
- Doesn’t need build information
- 223 CVEs described

What?

- CVEhound tool to detect absence of patches in the kernel sources
  - Project started in December 2020
  - GPLv3 for python code, GPLv2 for CVE detection rules
- Doesn’t look at kernel version
- Doesn’t require .git
- Doesn’t need build information
- 223 CVEs described
- Filters
  - CONFIG_ options
    - .config analysis based on undertaker project
  - Files, subdirectories
  - CWE, exploit status, …

What?

- CVEhound tool to detect absence of patches in the kernel sources
  - Project started in December 2020
  - GPLv3 for python code, GPLv2 for CVE detection rules
- Doesn’t look at kernel version
- Doesn’t require .git
- Doesn’t need build information
- 223 CVEs described
- Filters
  - CONFIG_ options
    - .config analysis based on undertaker project
  - Files, subdirectories
  - CWE, exploit status, …
- CVE metadata from linuxkernelcves.com
- Reports generation for CI

Why?

Context

- Kernel vulnerabilities
  - > 3000 CVE records on MITRE and NIST
  - > 1800 CVE records on linuxkernelcves.com
  - > 700 CVE records and ~50 BDU records on FSTEC BDU (since 2014)
  - > 1400 DWF/UVI records (since 2021)
  - …

Why?

Context

- Kernel vulnerabilities
  - > 3000 CVE records on MITRE and NIST
  - > 1800 CVE records on linuxkernelcves.com
  - > 700 CVE records and ~50 BDU records on FSTEC BDU (since 2014)
  - > 1400 DWF/UVI records (since 2021)
  - …
- Stable, LTS, XLTS, SLTS (CIP)
  - 4.4, 4.9, 4.19, 5.4, 5.10, 5.14
- Distributions
  - 3.10, 4.1, 4.18, 4.15, 5.3 …
- Embedded devices, mobile phones,…

Why?

Context

- Kernel vulnerabilities
  - > 3000 CVE records on MITRE and NIST
  - > 1800 CVE records on linuxkernelcves.com
  - > 700 CVE records and ~50 BDU records on FSTEC BDU (since 2014)
  - > 1400 DWF/UVI records (since 2021)
  - …
- Stable, LTS, XLTS, SLTS (CIP)
  - 4.4, 4.9, 4.19, 5.4, 5.10, 5.14
- Distributions
  - 3.10, 4.1, 4.18, 4.15, 5.3 …
- Embedded devices, mobile phones,…
- Many arches and CONFIG_* options
  - 17452 CONFIG_ options in v5.14

Why?

Context

- Kernel vulnerabilities
  - > 3000 CVE records on MITRE and NIST
  - > 1800 CVE records on linuxkernelcves.com
  - > 700 CVE records and ~50 BDU records on FSTEC BDU (since 2014)
  - > 1400 DWF/UVI records (since 2021)
  - …
- Stable, LTS, XLTS, SLTS (CIP)
  - 4.4, 4.9, 4.19, 5.4, 5.10, 5.14
- Distributions
  - 3.10, 4.1, 4.18, 4.15, 5.3 …
- Embedded devices, mobile phones,…
- Many arches and CONFIG_* options
  - 17452 CONFIG_ options in v5.14
- .git is not always available
  - only 427 commits with explicit CVE mentions (v5.14)
- …

Why?

Context

- Kernel vulnerabilities
  - > 3000 CVE records on MITRE and NIST
  - > 1800 CVE records on linuxkernelcves.com
  - > 700 CVE records and ~50 BDU records on FSTEC BDU (since 2014)
  - > 1400 DWF/UVI records (since 2021)
  - …
- Stable, LTS, XLTS, SLTS (CIP)
  - 4.4, 4.9, 4.19, 5.4, 5.10, 5.14
- Distributions
  - 3.10, 4.1, 4.18, 4.15, 5.3 …
- Embedded devices, mobile phones,…
- Many arches and CONFIG_* options
  - 17452 CONFIG_ options in v5.14
- .git is not always available
  - only 427 commits with explicit CVE mentions (v5.14)
- …

Tool use cases

- Certification Lab/Pentest Lab
  - Check all CVEs fixed for certification
  - Find unfixed CVEs to further check how they are mitigated with hardening options
- Users/System Administrators
  - Check kernels when you can't update it
  - Check sources before enabling kernel options
- Kernel developers
  - another tool to check yourself (reverts, wrong backports, early versions of patches)

How?

- Detection rules
  - Coccinelle patterns

```
// $ spatch rule.cocci .
@@
expression E;
@@

* copy_from_user(E, ...)
  ... when != E
* \(strncmp\|strcmp\)(..., E, ...)
```

How?

- Detection rules
  - Coccinelle patterns

```
// drivers/remoteproc/remoteproc_cdev.c
ret = copy_from_user(cmd, buf, len);
if (ret)
    return -EFAULT;

if (!strncmp(cmd, "start", len)) {
    if (rproc->state == RPROC_RUNNING ||
        rproc->state == RPROC_ATTACHED)
            return -EBUSY;
```

```
// net/core/pktgen.c
if (copy_from_user(f, &user_buffer[i], len))
    return -EFAULT;
i += len;

if (strcmp(f, "start_xmit") == 0) {
    pkt_dev->xmit_mode = M_START_XMIT;
} else if (strcmp(f, "netif_receive") == 0) {
```

Generic

```
// $ spatch rule.cocci .
@@
expression E;
@@

* copy_from_user(E, ...)
  ... when != E
* \(strncmp\|strcmp\)(..., E, ...)
```

```
// drivers/staging/rtl8723bs/os_dep/ioctl_linux.c
if (copy_from_user(new_ifname, wrqu->data.pointer, IFNAMSIZ))
    return -EFAULT;

if (0 == strcmp(rereg_priv->old_ifname, new_ifname))
        return ret;
```

...

How?

- Detection rules
  - Coccinelle patterns

Generic

```
// $ spatch rule.cocci .
@@
expression E;
@@

* copy_from_user(E, ...)
  ... when != E
* \(strncmp\|strcmp\)(..., E, ...)
```

```
// drivers/remoteproc/remoteproc_cdev.c
ret = copy_from_user(cmd, buf, len);
if (ret)
    return -EFAULT;

if (!strncmp(cmd, "start", len)) {
    if (rproc->state == RPROC_RUNNING ||
        rproc->state == RPROC_ATTACHED)
            return -EBUSY;
```

```
// net/core/pktgen.c
if (copy_from_user(f, &user_buffer[i], len))
    return -EFAULT;
i += len;

if (strcmp(f, "start_xmit") == 0) {
    pkt_dev->xmit_mode = M_START_XMIT;
} else if (strcmp(f, "netif_receive") == 0) {
```

```
// drivers/staging/rtl8723bs/os_dep/ioctl_linux.c
if (copy_from_user(new_ifname, wrqu->data.pointer, IFNAMSIZ))
    return -EFAULT;

if (0 == strcmp(rereg_priv->old_ifname, new_ifname))
        return ret;
```

...

How?

- Detection rules
  - Coccinelle patterns

Generic

```
// $ spatch rule.cocci .
@@
expression E;
@@

* copy_from_user(E, ...)
  ... when != E
* \(strncmp\|strcmp\)(..., E, ...)
```

```
// drivers/remoteproc/remoteproc_cdev.c
ret = copy_from_user(cmd, buf, len);
if (ret)
    return -EFAULT;

if (!strncmp(cmd, "start", len)) {
    if (rproc->state == RPROC_RUNNING ||
        rproc->state == RPROC_ATTACHED)
            return -EBUSY;
```

```
// net/core/pktgen.c
if (copy_from_user(f, &user_buffer[i], len))
    return -EFAULT;
i += len;

if (strcmp(f, "start_xmit") == 0) {
    pkt_dev->xmit_mode = M_START_XMIT;
} else if (strcmp(f, "netif_receive") == 0) {
```

```
// drivers/staging/rtl8723bs/os_dep/ioctl_linux.c
if (copy_from_user(new_ifname, wrqu->data.pointer, IFNAMSIZ))
    return -EFAULT;

if (0 == strcmp(rereg_priv->old_ifname, new_ifname))
        return ret;
```

...

How?

- Detection rules
  - Coccinelle patterns

~~Generic~~

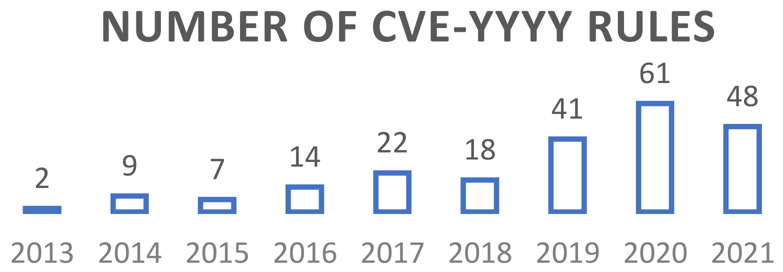
```
// $ spatch rule.cocci .
@@
expression E;
@@
pktgen_if_write(...)
{
    ... when any
*   copy_from_user(E, ...)
    ... when != E
*   strcmp(E, "start_xmit", ...)
    ...
}
```

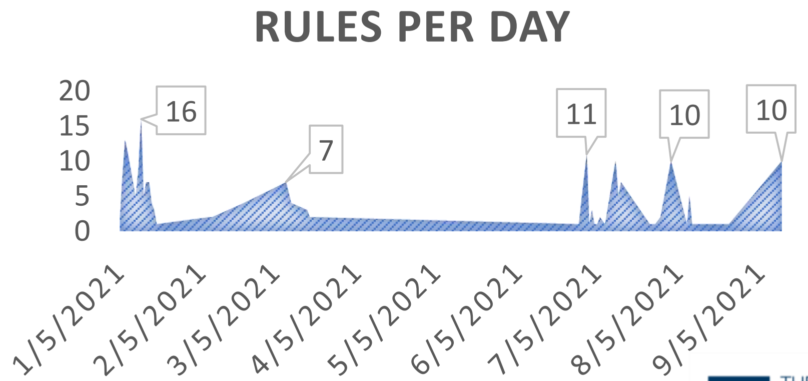
```
// net/core/pktgen.c
if (copy_from_user(f, &user_buffer[i], len))
    return -EFAULT;
i += len;

if (strcmp(f, "start_xmit") == 0) {
    pkt_dev->xmit_mode = M_START_XMIT;
```

How?

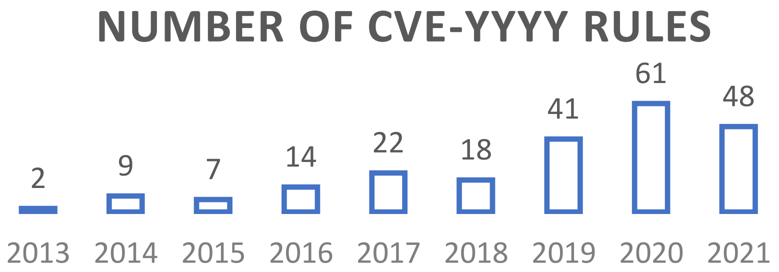
- Detection rules
  - Coccinelle patterns
- Stats
  - since December 2020
  - 42 days with new rules
  - ~5 rules a day
  - only 8 days with >=10 rules

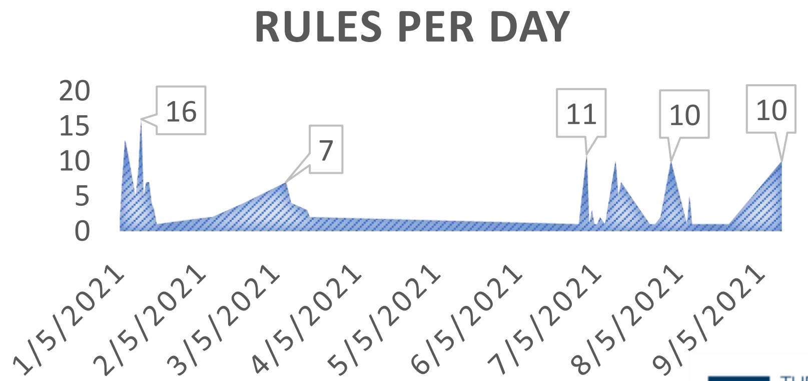




How?

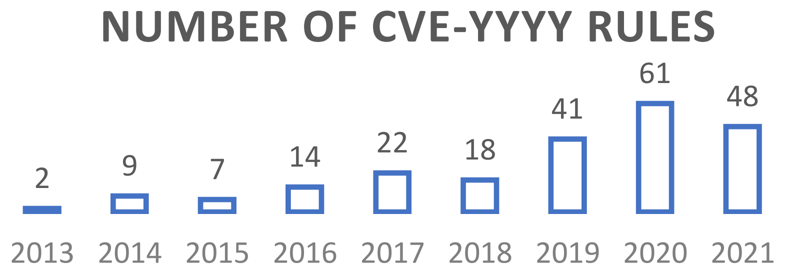
- Detection rules
  - Coccinelle patterns
- Stats
  - since December 2020
  - 42 days with new rules
  - ~5 rules a day
  - only 8 days with >=10 rules
- Testing
  - Detects between [fixes, fix) commits
  - Not detects on v2.6.12-rc2, fixes^, master, stable/linux-d.dd.y, next/master

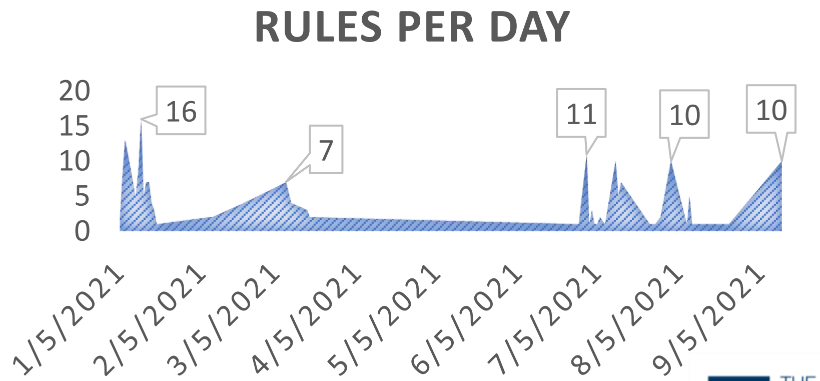




How?

- Detection rules
  - Coccinelle patterns
- Stats
  - since December 2020
  - 42 days with new rules
  - ~5 rules a day
  - only 8 days with >=10 rules
- Testing
  - Detects between [fixes, fix) commits
  - Not detects on v2.6.12-rc2, fixes^, master, stable/linux-d.dd.y, next/master
- How to write a rule
  1. Find fix commit
  2. Find fixes commit (only for testing)
  3. Draft the rule (5-30 mins)
  4. Test the rule (7-20 mins)
  5. Refine the rule (repeat to 4)





Rule patterns (removed, added)

CVE-2020-28097

```
@err exists@
position p;
@@

vgacon_scrollback_init@p(...)
{
    ... when any
    CONFIG_VGACON_SOFT_SCROLLBACK_SIZE
    ... when any
}

@script:python@
p << err.p;
@@
coccilib.report.print_report(p[0],
       'ERROR: CVE-2020-28097')
```

CVE-2020-26088

```
@err exists@
identifier sock;
position p;
@@

rawsock_create(...)
{
       ... when != if (!\(ns_capable\|capable\)
                    (..., CAP_NET_RAW)) return -EPERM;
       sock->ops =@p &rawsock_raw_ops;
       ...
}
@script:python@
p << err.p;
@@
coccilib.report.print_report(p[0],
                            "ERROR: CVE-2020-26088")
```

Rule patterns (added alt., changed)

CVE-2020-27068

```
@enum_status_code@
@@
enum nl80211_attrs {
        ...,
        NL80211_ATTR_STATUS_CODE,
        ...
};
@fix@
@@
struct nla_policy nl80211_policy[...] = {
        ...,
        [NL80211_ATTR_STATUS_CODE] = { ... },
        ...
};
@err depends on enum_status_code && !fix@
position p;
@@
struct nla_policy nl80211_policy@p[...] = { ... };
```

CVE-2020-12912

```
@err@
position p;
@@

amd_energy_is_visible(...)
{
        return 0444;@p
}
```

Rule patterns (fix != bug)

CVE-2020-28941

```
@close exists@
@@
spk_ttyio_ldisc_close(...) {
        ...
        kfree(speakup_tty->disc_data);
        ...
}
@err depends on close exists@
@@
spk_ttyio_ldisc_open(...) {
        ... when != mutex_lock(...);
        speakup_tty = tty;
        ...
        speakup_tty->disc_data = ldisc_data;
        ... when != mutex_unlock(...);
}
```

CVE-2018-20855

```
@struct_fields@
@@
struct mlx5_ib_create_qp_resp {
        __u32 bfreg_index;
        ...
        __u32 reserved;
        ...
};
@err depends on struct_fields exists@
@@
create_qp_common(...) {
        ...
        struct mlx5_ib_create_qp_resp resp;
        ... when != memset(&resp, 0, ...)
        create_user_qp(..., &resp, ...)
        ...
}
```

THE LINUX FOUNDATION

Rule patterns (non-C ld, .S)

CVE-2021-3411

```
@initialize:python@
@@
int3 = False
with open(vmlinux_lds, 'rt') as f:
        if re.search(':text\s*=\s*0xcccc', f.read()):
                int3 = True
@err exists@
expression E;
@@
can_optimize(...) {
        ...
        if (E.opcode.bytes[0] ==
\(INT3_INSN_OPCODE\|BREAKPOINT_INSTRUCTION\))
                return 0;
        ...
}
@script:python@
p << err.p;
@@
if int3:
        coccilib.report.print_report(p[0], 'ERROR: CVE-2021-3411')
```

CVE-2017-1000255

```
// Fallback mode with regular expressions.
// grep -rePoz <regex1> && grep -rePoz <regex2>

tm_enabled\(struct\s+task_struct\s+\*[\w]+\)\s+\{

EXC_COMMON_BEGIN\(program_check_common\)\s+EXCEPTIO
N_PROLOG_COMMON\(0x700,\s+PACA_EXGEN\)
```

Future Work

- KernelCI integration
- Support #ifdefs
  - More precise CONFIG_* analysis
- Add option to check only specific drivers enabled by CONFIG_* option
- Lightweight mode based on .git analysis
  - Check for Fixes and Fix commits
  - Based on commits titles and commits metadata
  - Reverts, multiple commits fixing one CVE, …
  - Useful primarily for kernel developers
- Infer detection rules from commits

LINUX SECURITY SUMMIT

Just patch/patch -R!

- Old kernels?
  - Let’s try LTS patches
- What it means if a patch doesn’t apply/revert?
  - Already patched kernel with new changes on top of it?
  - Old non-vulnerable kernel?
  - There is no suitable backport of a patch to test?
  - Multiple patches fixing one CVE?
- What it means if a patch applies?
  - Fix != Error
  - Check patches that introduce errors?
- …