CVEs are alive, but do not panic!

Greg Kroah-Hartman

gregkh@linuxfoundation.org
git.sr.ht/~gregkh/presentation-security

All of this is just my personal opinion, based on working as part of the Linux kernel security team since it was created in 2005.

Nothing in here reflects the opinion of the Linux Foundation or any other Linux kernel developer.

But hopefully I can convince them to agree with me.

CVEs used to mean nothing for Linux

https://kernel-recipes.org/en/2019/talks/cves-are-dead-long-live-the-cve/

CVEs now mean something!

https://www.cve.org/Media/News/item/news/2024/02/13/kernel-org-Added-as-CNA

http://www.kroah.com/log/blog/2024/02/13/linux-is-a-cna/

kernel.org is now a CNA!

- Responsible for all kernel CVEs
- Community is in control
- CVEs assigned for all “vulnerabilities” fixed

Linux CVE resources

- Contact us:

  cve@kernel.org

- Process documentation:

  https://docs.kernel.org/process/cve.html

- Public git repo:

  https://git.kernel.org/pub/scm/linux/security/vulns.git/

- List of all assigned CVEs:

  https://lore.kernel.org/linux-cve-announce/

What is a vulnerability?

"An instance of one or more weaknesses in product that can be exploited, causing a negative impact to confidentiality, integrity, or availability; a set of conditions or behaviors that allows the violation of an explicit or implicit security policy."

– cve.org

What is a kernel vulnerability?

- Any user-triggerable crash or reboot
- Memory use-after-free / leak / overflow
- Incorrect boundary checks
- Denial of service
- Logic errors
- Lots of other things

Why is triggering `WARN_ON` a CVE?

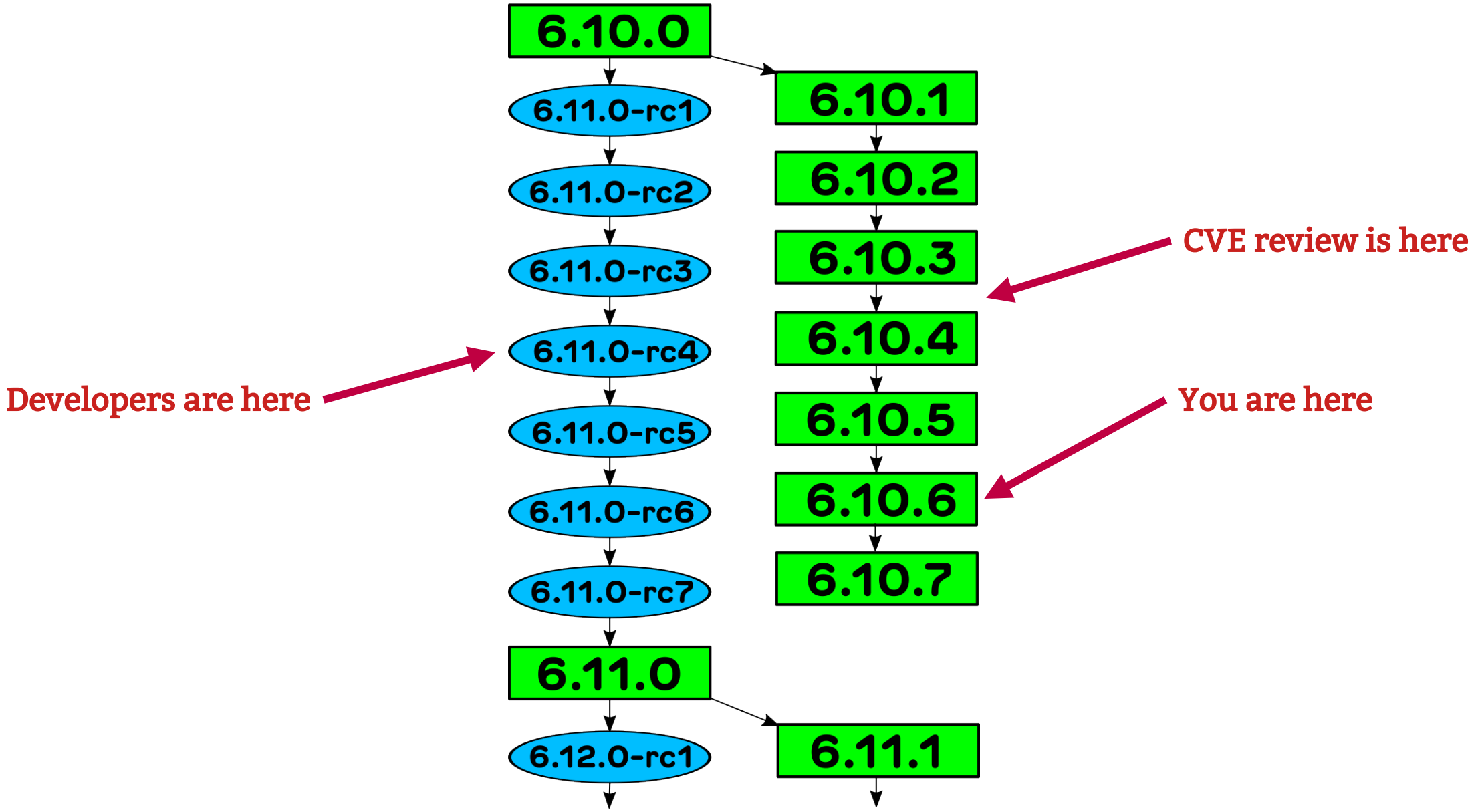
- If `panic_on_warn` is enabled, reboot happens
- Billions of Linux systems have this enabled
- Android is slowly moving away from this.
- Some want any warning to reboot

What is a not a kernel vulnerability?

- Data corruption / loss
- Performance issues
- Bugfixes that are not externally triggered

CVEs are assigned after-the-fact

- Usually 1-2 week delay
- Allows systems to be updated before public announcements
- CVEs only reference specific fix, not if any previous changes needed.
- CVE fixes are NOT tested independently


6.10.0
6.11.0-rc1
6.11.0-rc2
6.11.0-rc3
6.11.0-rc4
6.11.0-rc5
6.11.0-rc6
6.11.0-rc7
6.11.0
6.12.0-rc1
6.10.1
6.10.2
6.10.3
6.10.4
6.10.5
6.10.6
6.10.7
6.11.1
Developers are here
CVE review is here
You are here

No hardware CVEs

- SPECTRE-like CVEs come from others

No hardware CVEs

- SPECTRE-like CVEs come from others
- Good luck figuring this out

How are they assigned

- CVE team reviews every stable commit
- Votes if is, or is not, a vulnerability
- Each team member uses different methods
- Commits that are agreed get assigned
- Commits that are disagreed are discussed
- Review happens in public

How are they assigned – cont.

- Developer / community requests
- Corporate requests

Lots of CVEs!

- Averaging 55 CVEs per week

We are NOT #1!

- Wordpress – 112 per week
- MITRE – 95 per week
- Github – 37 per week
- Windows – 11-12 per week
- Chrome – 6 per week
- macOS – 6 per week
- iOS – 2-3 per week

All CVEs have increased

“We probably can fix this, if we want to, it’s pretty important, but so far we’ve successfully managed to ignore it. It’s not going to be easy. By definition, the people who got us here can’t fix this problem. They would have long ago if they could. Many of the people involved in the vulnerability space, CVE, NVD – they have been doing this work for decades. A bunch don’t think there’s anything wrong. If you talk to any vulnerability analyst, developer, or operations person actually working with CVE IDs, they’re miserable. They get their work done in spite of CVE IDs, not because of them.”

– Kurt Seifried & Josh Bressers

https://opensourcesecurity.io/2024/06/03/why-are-vulnerabilities-out-of-control-in-2024/

Kernel CVEs are descriptive

- Every CVE says what files are affected
- Every CVE says what versions are affected
- Very few CVEs are applicable for you!
- json format

`https://git.kernel.org/pub/scm/linux/security/vulns.git/`

`https://github.com/CVEProject/cvelistV5.git`

Most CVEs are not for you!

- Only you know your usecase!
- Only you know what files you use!

Some CVEs are for you!

- Real issues get fixed every week
- Ignoring them will affect your security

Staying secure

- Take all stable/longterm updates
- Community supported and tested as a whole
- Bonus is you get data corruption fixes and performance improvements for free!

“If you are not using the latest a stable / longterm kernel, your system is insecure”

– me

“In order to run the safest kernel, one must choose to either update to latest stables and do all the work needed to get it running in production, or one must triage every fix to find only those that need to be applied based on one's threat model.

Either way is likely a lot of work, but one must figure out which is the least amount of work, and then do it.

There isn't another path to running a kernel with the flaws fixed.”

– Kees Cook

2 options for not taking stable

- Triage all 55/week to determine what to take

2 options for not taking stable

- Triage all 55/week to determine what to take
- Fix your process to be able to take updates

Android example

- 6.6.49
  - 94 commits
  - 19 CVEs

Android example – cont.

- 6.6.49
  - 19 commits affect Android GKI build (3.6 million lines in build, no drivers)
  - 4 commits have CVEs assigned

Android example – cont.

- 6.6.49

```
4ed45fe99ec9 pinctrl: single: fix potential NULL dereference in pcs_get_function()
459584258d47 selinux,smack: don't bypass permissions check in inode_setsecctx hook
94ab317024ba ethtool: check device is present when getting link settings
7bb11a75dd4d usb: dwc3: core: Prevent USB core invalid event buffer address access

 drivers/pinctrl/pinctrl-single.c |    2 ++
 drivers/usb/dwc3/core.c          |    8 ++++++++
 net/core/net-sysfs.c             |    2 +-
 net/ethtool/ioctl.c              |    3 +++
 security/selinux/hooks.c         |    4 ++--
 security/smack/smack_lsm.c       |    4 ++--
 6 files changed, 18 insertions(+), 5 deletions(-)
```

Tools

- dyad – finds vulnerable/fixed git ranges
- digs across stable branches

```
$ dyad 371a3bc79c11
# dyad version: de9a7e3de532
#   getting vulnerable:fixed pairs for git id 371a3bc79c11b707d7a1b7a2c938dc3cc042fffb
4.13:349d39dc57396e3e9f39170905ff8d626eea9e44:4.14.189:39e0651cac9c80865b2838f297f95ffc0f34a1d8
4.13:349d39dc57396e3e9f39170905ff8d626eea9e44:4.19.134:febe56f21371ba1e51e8586c3ddf8f54fc62fe61
4.13:349d39dc57396e3e9f39170905ff8d626eea9e44:5.4.53:d3b7bacd1115400b94482dfc7efffc175c29b831
4.13:349d39dc57396e3e9f39170905ff8d626eea9e44:5.7.8:9006b543384ab10902819364c1205f11a1458571
4.13:349d39dc57396e3e9f39170905ff8d626eea9e44:5.8:371a3bc79c11b707d7a1b7a2c938dc3cc042fffb
```

Our commits are a mess

- Invalid git ids everywhere
- Invalid or incorrect“Fixes:” tags
- Multiple “Fixes:” tags across releases
- Fixes going backwards in time
- Vulnerable only in branches
- No fixes despite the tag saying so
- …

Tools

- bippy – creates cve entries for a git commit

```
Usage: ./bippy [OPTIONS]
Create a JSON or MBOX file to report a CVE based on a specific Linux kernel git sha value.

Arguments:
 -c, --cve=CVE_NUMBER      The full CVE number to assign
 -s, --sha=GIT_SHA         The kernel git sha1 to assign the CVE to
 --vulnerable=GIT_SHA      The kernel git sha1 that this issue became vulnerable at. (optional)
 -j, --json=JSON_FILENAME  Output a JSON report to submit to CVE to the
                           specified filename
 -m, --mbox=MBOX_FILENAME  Output a mbox file to submit to the CVE
                           announce mailing list
 --diff=DIFF_FILENAME      File containing a diff for the changelog text to be applied. (optional)
 --reference=REFERENCE_FILENAME File containing a list of url references to add to the json record.
(optional)
 -u, --user=EMAIL          Email of user creating the record.
 -n, --name=NAME           Name of the user creating the record.
 -h, --help                This information
 -v, --verbose             Show debugging information to stdout

Note, CVE_NUMBER and GIT_SHA are required, as well as at least one of
JSON_FILENAME and/or MBOX_FILENAME.
If EMAIL or NAME is not specified, they will be taken from 'git config' user settings.
```

Tools

- strak – tool to show how vulnerable you are

```
$ strak v6.10.11
v6.10.11 is vulnerable to CVE-2024-41013
v6.10.11 is vulnerable to CVE-2024-41014
v6.10.11 is vulnerable to CVE-2024-41016

$ strak --fixed 6.6.49
CVE-2024-43891 is fixed in 6.6.49 with commit 4ed03758ddf0
CVE-2024-46673 is fixed in 6.6.49 with commit 8a3995a3ffec
CVE-2024-46674 is fixed in 6.6.49 with commit e1e5e8ea2731
CVE-2024-46675 is fixed in 6.6.49 with commit 7bb11a75dd4d
CVE-2024-46676 is fixed in 6.6.49 with commit 56ad559cf6d8
CVE-2024-46677 is fixed in 6.6.49 with commit 28c67f0f84f8
CVE-2024-46678 is fixed in 6.6.49 with commit 6b598069164a
CVE-2024-46679 is fixed in 6.6.49 with commit 94ab317024ba
CVE-2024-46680 is fixed in 6.6.49 with commit 662a55986b88
CVE-2024-46685 is fixed in 6.6.49 with commit 4ed45fe99ec9
CVE-2024-46686 is fixed in 6.6.49 with commit a01859dd6aeb
CVE-2024-46687 is fixed in 6.6.49 with commit 51722b99f41f
```

“In order to run the safest kernel, one must choose to either update to latest stables and do all the work needed to get it running in production, or one must triage every fix to find only those that need to be applied based on one's threat model.

Either way is likely a lot of work, but one must figure out which is the least amount of work, and then do it.

There isn't another path to running a kernel with the flaws fixed.”

– Kees Cook

CVEs are alive, but do not panic!

Greg Kroah-Hartman

gregkh@linuxfoundation.org
git.sr.ht/~gregkh/presentation-security