Kernel hacking behind closed doors

Thomas Gleixner

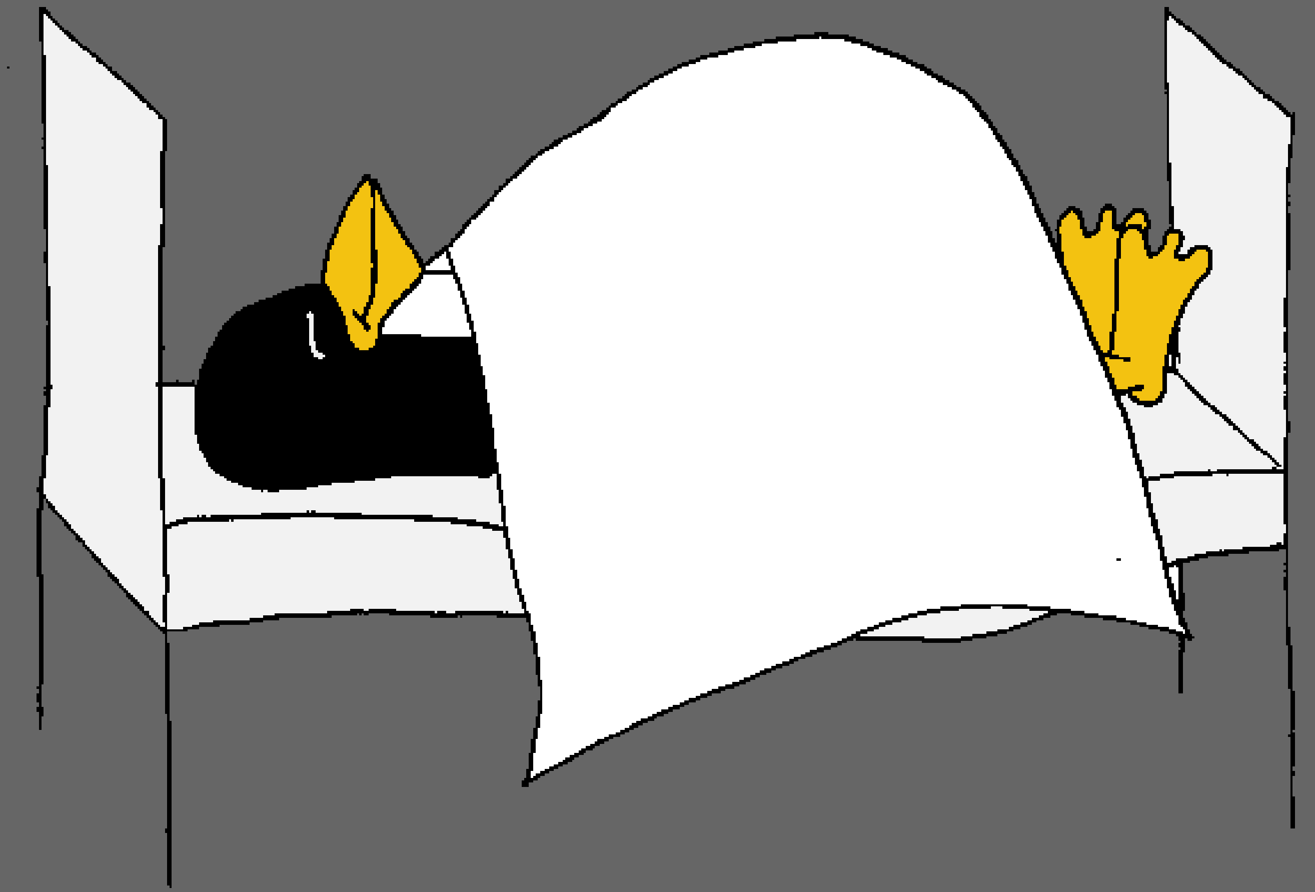
Kernel-Recipes 2019

Software security bugs

Software security bugs

- Affect usually only the Linux kernel.
- Security team can freely decide whom to bring in for addressing the issue.
- Aims for disclosure of fixes as soon as they are available.
- Coordinated release with a 7 days embargo, in exceptional cases 14 days.

For a long time …



Do we really have to deal with this?

Hardware security bugs

Hardware security bugs

- Affect all OS vendors / projects
- Disclosure process to kernel developers is sensitive
- Disclosure of fixes needs industry wide coordination
- Potentially long embargo times due to dependency on firmware, microcode updates

Software vs. hardware security bugs

Software security bugs

- Affect usually only the Linux kernel.
- Security team can freely decide whom to bring in for addressing the issue.
- Disclosure of fixes as soon as they are available.
- Coordinated release with a 7 days embargo, in exceptional cases 14 days.

Hardware security bugs

- Affect all OS vendors / projects
- Disclosure process to kernel developers is sensitive
- Disclosure of fixes needs industry wide coordination
- Potentially long embargo times due to dependency on firmware, microcode updates

2017

07/17	• Several research teams discover Meltdown/Spectre • Intel is disclosed and takes over coordination
08/17	• Microsoft, Apple and others are disclosed • „Don‘t worry about Linux, we are taking care of that“
09/17	• RedHat, Suse and other vendors are disclosed, but can‘t talk to each other • Intel has separate teams for each vendor
10/17	• Partial Meltdown disclosure to x86 maintainers • Kaiser patches are posted
11/17	• Microsoft releases Beta version with KPTI (discovered by Alex Ionescu) • Kaiser patches are cleaned up and renamed to PTI
12/17	• KPTI is gradually merged • On Dec. 21th, Intel discloses Spectre to x86 maintainers
01/18	• On Jan. 3rd, Meltdown/Spectre goes public five days early

Panic engineering

- Three different patch sets addressing Spectre
- All broken
- Discussion is leading nowhere due to lack of information
- Five people eight opinions

Panic engineering

```
> +.macro WRMSR_ASM msr_nr:req edx_val:req eax_val:req
> +     movl    \msr_nr, %ecx
> +     movl    \edx_val, %edx
> +     movl    \eax_val, %eax
> +.endm
```

This is the most brilliant piece of useless code I've seen in a long time.

Panic engineering

> Rather than continuing to debate it, perhaps it's best just to wait for
> the US to wake up, and Intel to give a definitive answer.

So here is the simple list of questions all to be answered with YES or NO.

I don't want to see any of the 'but, though ...'. We all know by now that it's CPU dependent and slow and whatever and that IBRS_ATT will be in future CPUs.

So get your act together and tell a clear YES or NO.

Going back to normal

After 10 days of frenzy following the disclosure of the mess, I'm at a point where I think that we have reached a state where the main targets are covered...

We all are exhausted and at our limits and I think we can agree that having the most problematic stuff covered is the right point to calm down and put the heads back on the chickens. Take a break and have a few drinks at least over the weekend!

To be honest the last 10 days were more horrible than the whole PTI work due to lack of documentation, 12 different opinions when asking 8 people (why does this have a lawyer smell?) and an amazing amount of half baken and hastily cobbled together crap.

Please let‘s stop this and return to normality now.

Agreement between community and vendors

- Industry wide collaboration
- No compartementation
- Full information disclosure is required
- Upstream first

How to communicate securely?

- Keybase IO ?
- Maintain CC lists and PGP encrypt everything ?
- Not only Linus hates PGP for good reasons
- Encrypted mailing-lists to the rescue

Encrypted mailinglist

- Only a few projects
- Some are abandoned
- One is S/MIME only
- One is PGP only

Encrypted mailinglist - Test

- Install on your mail server !?!
- Got it „running“ in a secured VM on a secured host
- Three out of ten emails break it
- List engine in the middle of a maze of webservices, mail transport, SQL and other unpenetratable Ruby code
- What now?

Encrypted mailinglist - DIY

- Python to the rescue
- Trivial pipe based script, no mail transport except SMTP to localhost. Getmail just works (most of the time)
- Yaml based configuration managed via git
- S/MIME + PGP in and out
- Python2 email handling sucks
- Three days later ….

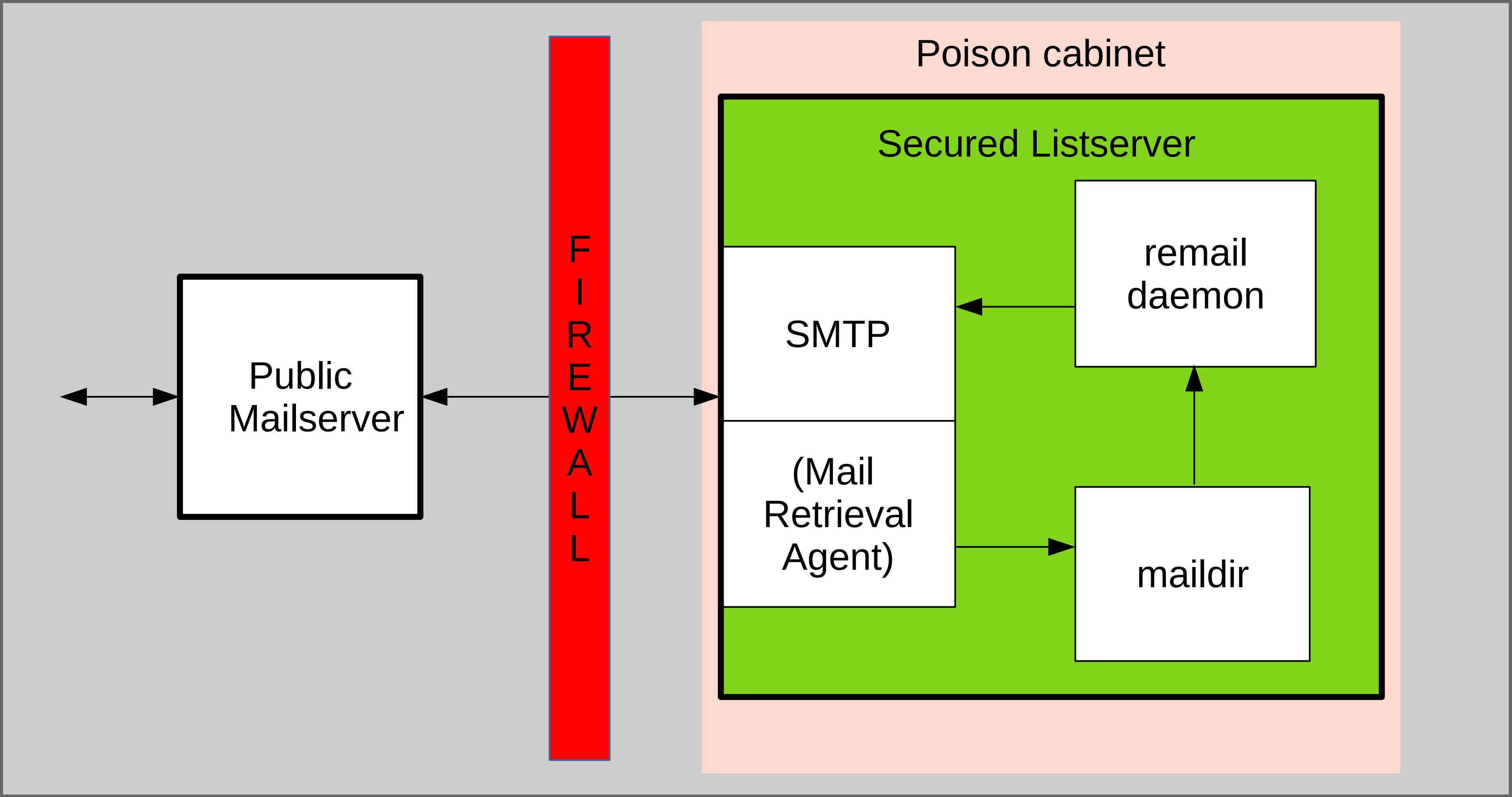
Encrypted mailinglist - DIY

- Break a few corporate mail servers
- Deal with all sorts of mail clients
- Distangle the all in one script into proper python
- Switch from pipe to maildir
- Convert to Python3

After 18 month and more than 3500 incoming mails handled:

https://git.kernel.org/pub/scm/linux/kernel/git/tglx/remail.git

Encrypted mailinglist - DIY



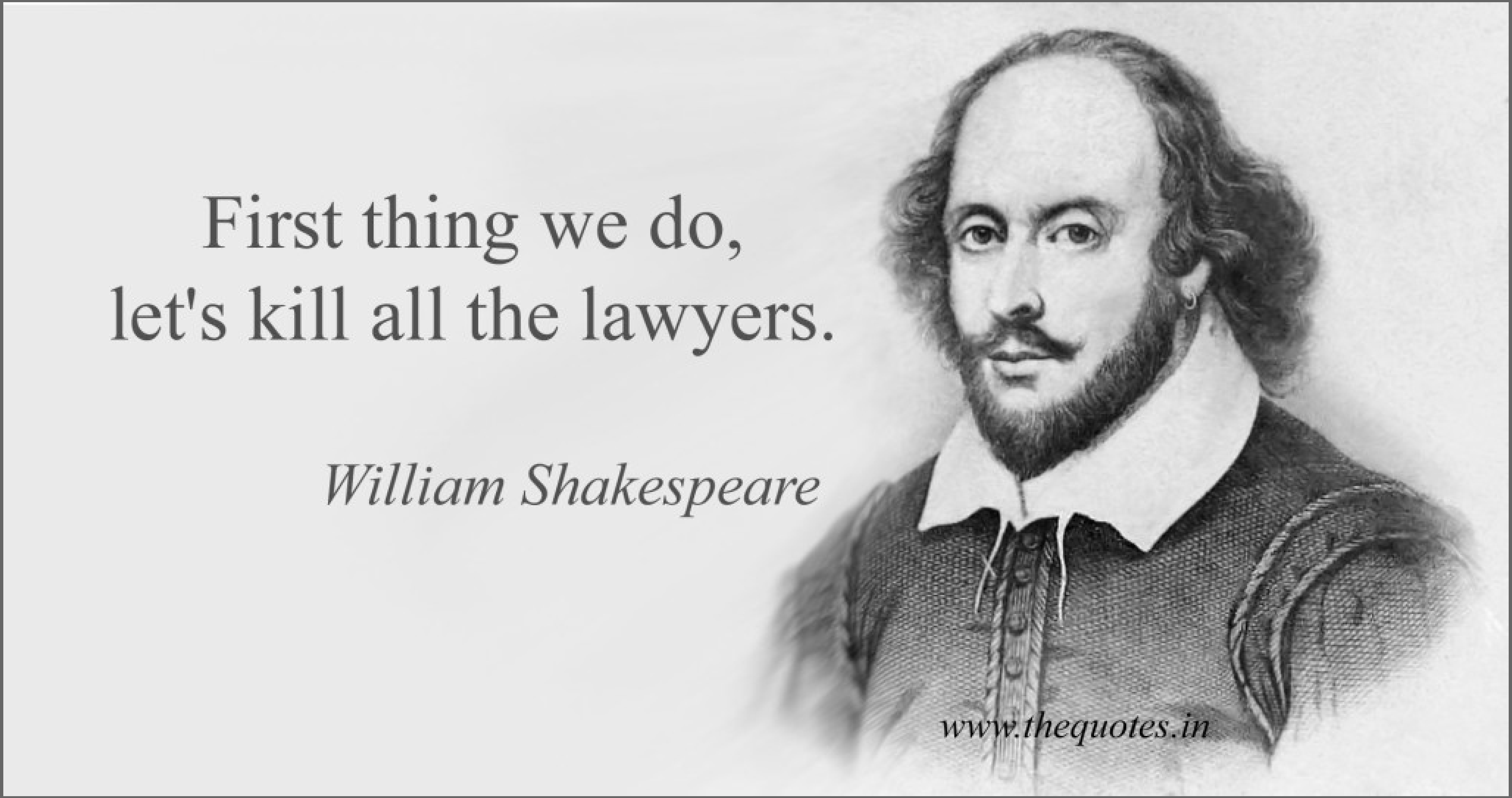
SSB, L1TF, MDS

- Working together
- Upstream first aligns community and vendors
- Almost business as usual
- Still there is a problem

The problem

- Bringing in experts is painful
- The disclosure and handling is controlled by lawyers
- The universal lawyer tool for this are NDAs
- NDAs are not workable as the Kernel community is not a formal body
- What now?

Solution #1

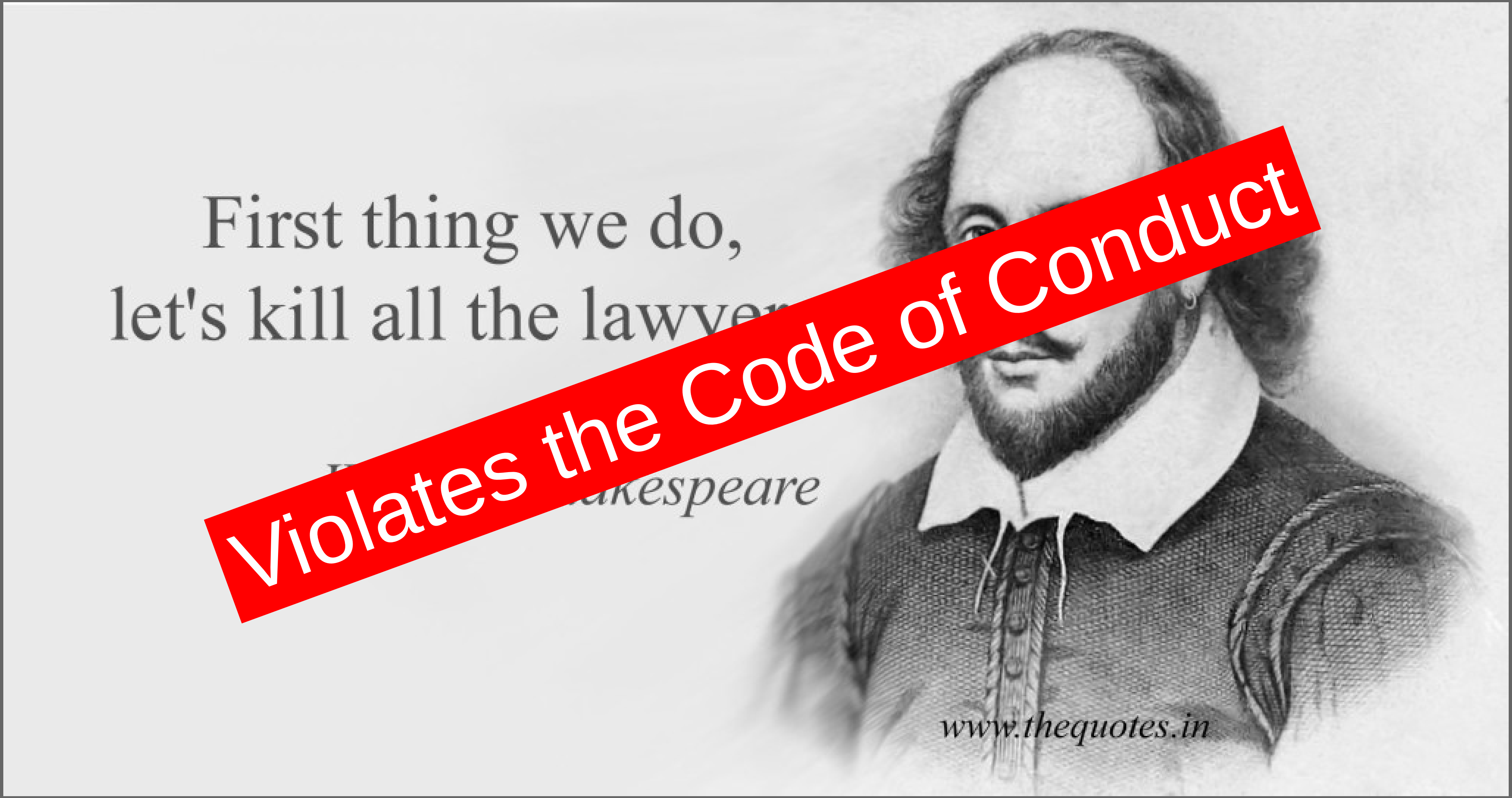


First thing we do,
let's kill all the lawyers.

William Shakespeare

www.thequotes.in

~~Solution #1~~



Violates the Code of Conduct

Solution #2

- Provide a formal process
- Provide a NDA substitute

Formal process

- Separate point of contact
  - Solely for Hardware security issues
  - Published list of security officers
  - Encrypted mailing lists
- Strict disclosure rules for kernel developers (domain experts). Aware of potential conflicts of interest
- Disclosed kernel developers adhere to a Memorandum of Understanding
- Embargo / disclosure coordination accross the industry

See: https://www.kernel.org/doc/html/latest/process/embargoed-hardware-issues.html

Memorandum Of Understanding

The Linux kernel community has a deep understanding of the requirement to keep hardware security issues under embargo for coordination between different OS vendors, distributors, hardware vendors and other parties.

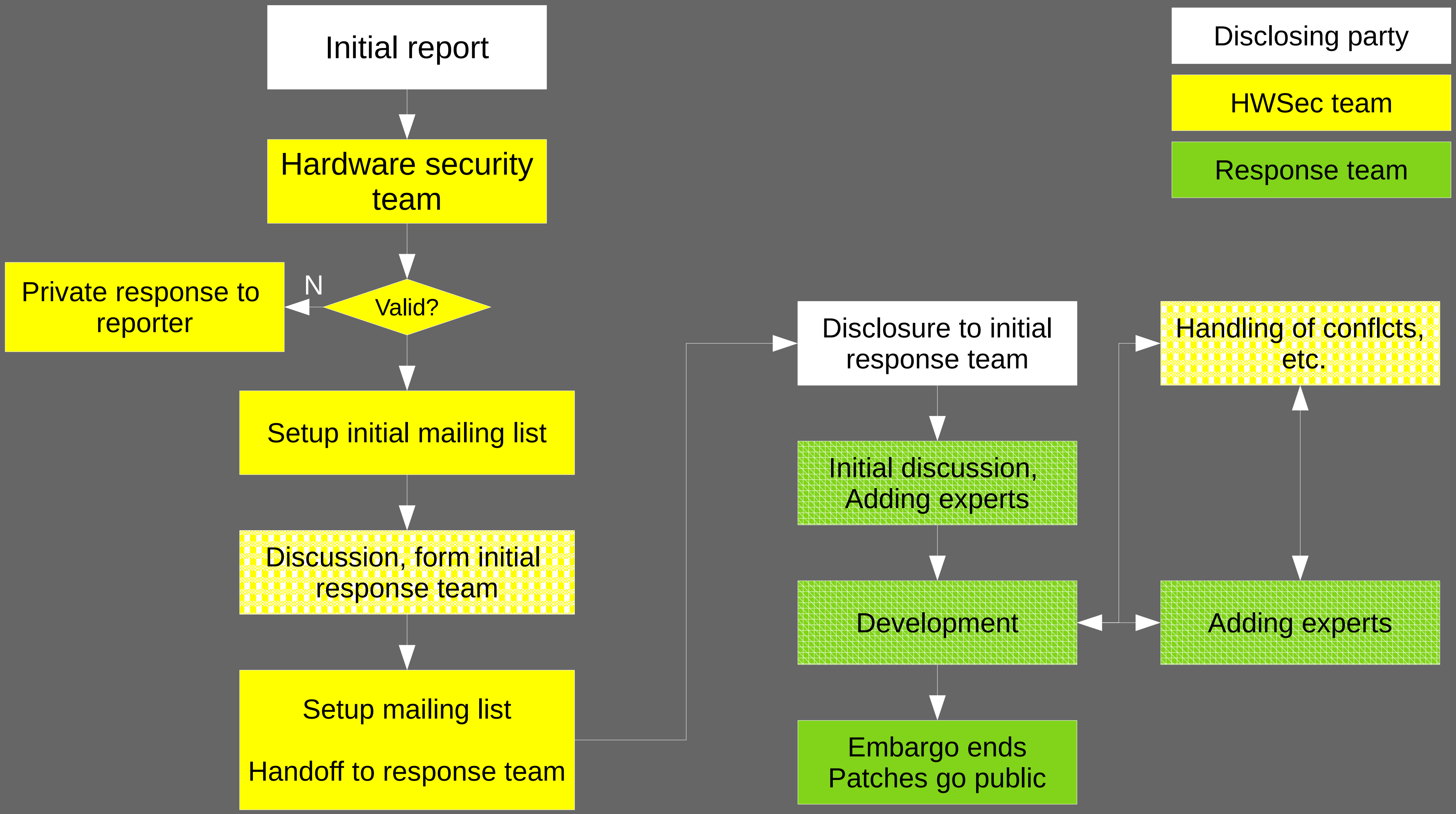
The Linux kernel community has successfully handled hardware security issues in the past and has the necessary mechanisms in place to allow community compliant development under embargo restrictions.

The Linux kernel community has a dedicated hardware security team for initial contact, which oversees the process of handling such issues under embargo rules.

The hardware security team identifies the developers (domain experts) who will form the initial response team for a particular issue. The initial response team can bring in further developers (domain experts) to address the issue in the best technical way.

All involved developers pledge to adhere to the embargo rules and to keep the received information confidential. Violation of the pledge will lead to immediate exclusion from the current issue and removal from all related mailing-lists. In addition, the hardware security team will also exclude the offender from future issues. The impact of this consequence is a highly effective deterrent in our community. In case a violation happens the hardware security team will inform the involved parties immediately. If you or anyone becomes aware of a potential violation, please report it immediately to the Hardware security officers.

Formal process


Initial report
Hardware security team
Valid?
N
Private response to reporter
Setup initial mailing list
Discussion, form initial response team
Setup mailing list
Handoff to response team
Disclosure to initial response team
Initial discussion, Adding experts
Development
Embargo ends Patches go public
Handling of conflcts, etc.
Adding experts
Disclosing party
HWSec team
Response team

Industry acceptance

- All major players agreed
- Established process embassadors
- But ...

Hopefully we won‘t ever need that again.
At least not before I retired.

Thomas Gleixner 2019