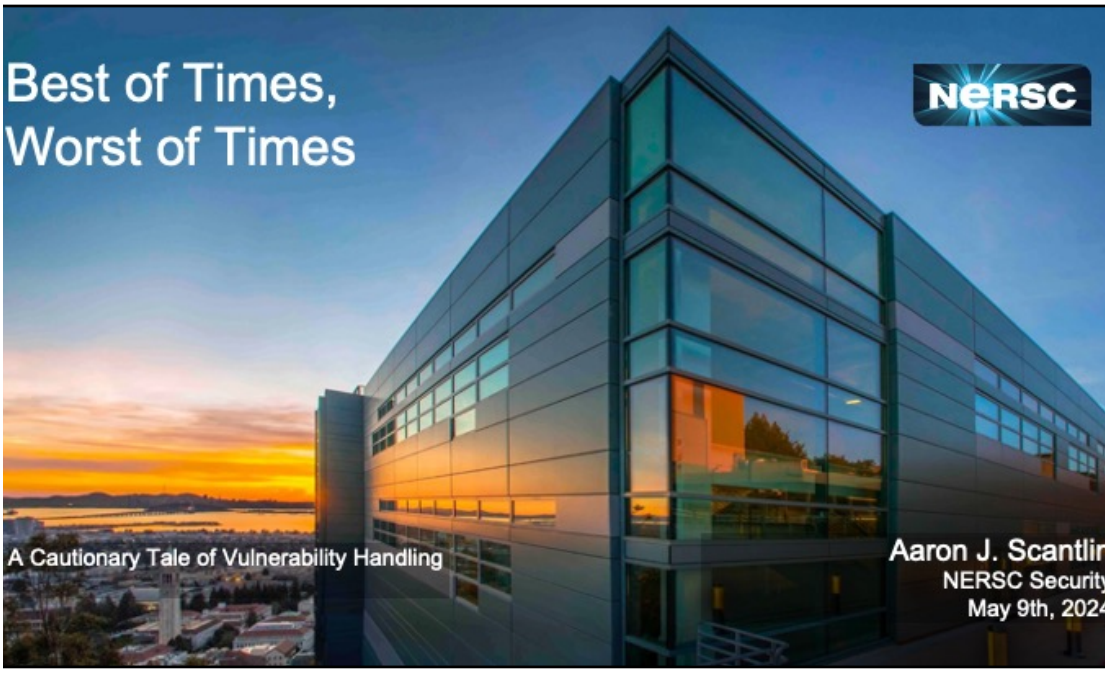


Good morning everyone! I've got a lot to cover, so hold onto your butts - I'm gonna move fast, but I'm hoping that will leave us a bit more time for discussion at the end. We'll see how it goes. :)

Also: if you're reviewing these slides after-the-fact, please feel free to reach out if you have any questions or comments!
ascantlin@lbl.gov

Agenda

- Who's the weirdo talking to you
- Why's the weirdo talking to you
- Timeline covering discovery -> public patch availability
- The vulnerability in question (CVE-2023-51786)
- Major issues experienced during triage
- Thoughts on how to make this a less painful experience for our entire community
- Call to action
- Questions / comments / daps / high-fives / etc

I'll give folks a few moments to review the agenda and decide if they want to bug out to a different talk - no qualms here if so! Your time is valuable, and ah… well, as I've had the pleasure of listening to the other technical talks this week… I'm noticing that "infosec talks" seem to be a *bit* different from the "technical talks" y'all might be used to seeing. I have no doubt you'll find this content *entertaining* - I think you'll find more quips than graphs and charts - but perhaps that's to be somewhat expected given the subject matter.

Today, we'll discuss the technical details of CVE-2023-51786, as well as NERSC's experiences while discovering and reporting it. I will also propose what I believe may be a way to work toward ensuring other sites don't have to go through similar pains, beg – er, *politely request*, your support in making it a reality, and then open the floor to gauge whether what I'm proposing sounds like it may have some merit, or find out I'm barking up the wrong tree. :)

> whoami



- Aaron J. Scantlin
  - Cybersecurity Engineer at NERSC
    - Security practitioner for ~10 years, security nerd for ~15 years
  - Resident Zeek Geek
  - Member: SANS Advisory Board
  - Member: REN-ISAC
  - Member: SCinet (DevOps - and maybe Netsec?)
  - OG @ SecKC (https://www.seckc.org)
  - DEFCON Goon (NFO)
  - Master of Ceremonies @ BSidesKC
  - Lead Cat Herder in today's story
  - Neurodivergent, family man, hip-hop head, rhythm gamer

NERSC 3 BERKELEY LAB U.S. DEPARTMENT OF ENERGY Office of Science

Hopefully we all saw Wahid's award-winning presentation yesterday - and since you already know about NERSC, but probably have no clue who I am (first time CUG attendee), I thought I'd take a few moments to introduce myself. So, hi! Aaron Scantlin, Cybersecurity Engineer at NERSC; I'm a firm believer that you shouldn't listen to anyone tell you anything they ain't got no business tellin' you 'bout - and while I'm relatively new to HPC, I'm getting to be an oldhead in infosec. I was a security analyst and adjunct professor for the University of Missouri - Columbia campus prior to my current role; this has served me pretty well since open science enclaves have a security posture and risk appetite more closely resembling an R1 university than, say, Los Alamos National Lab - or as I like to call them, the "boom boom" Labs.

This is a list of things I am or do - hopefully illustrating I'm at least partially qualified to speak on today's topic! You may have noticed I have a tendency to stand alone or off to the side in crowds or groups - I sometimes have a hard time initiating

conversation if I don’t know somebody, so to those of you who have a
thank you. To those of you I haven’t had the pleasure of getting to me
anything else you see on this slide - I’ll put my contact info up at the er
(or if you’re like me and have a hard time doing that in meatspace.)

The Real Heros of This Story

Laurie Stephey
Brian Friesen
Lisa Gerhardt
Eric Roman
Chris Samuel
Rebecca Hartman-Baker
Daniel Fulton
Ravi Cheema
David Fox
Shane Canon

Before we get too far, I want to take a moment to shout out all the people who did the **real** work here.

My role in this was to herd cats and (begrudgingly) enforce the sometimes necessary clandestine nature of secops. The people you see on this slide were involved in the "meat-n-potatoes" of analyzing the root cause of the vulnerability, communication with impacted users, working with HPE to ensure the vuln was triaged appropriately, and/or implementing risk mitigations. I am quite sure I forgot some, so if you should have been on this list and aren't, please forgive me!

Motivation

~~spite, mostly~~

As for my motivation…

Motivation

- Reduce “the dumpster fire effect” of discovering vulnerabilities in research computing
- Provide specialized security support for a field traditionally less concerned* with security
- Accountability for vendors and researchers alike
- Have a way to protect each other without impacting responsible disclosure
- **Have vulnerability reports triaged by security professionals who can advocate for sites’ needs**

*not universal - but somewhat common in open science environments

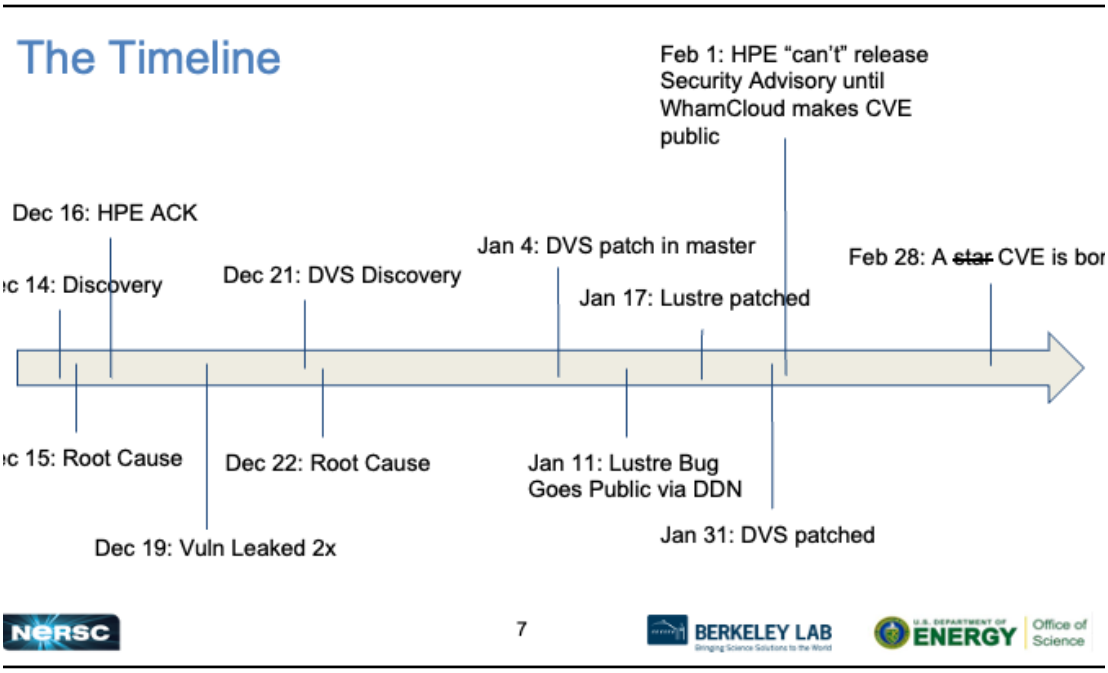
Honestly though - what we went through, in a word, SUCKED. I don’t ever wanna deal with that again if I can help it, and I sure as heck don’t wanna be on the “impacted but unaware” side that so many other sites found themselves on. I ah… honestly submitted this abstract out of frustration with how HPE handled the situation. So when I got the email from Bilel saying my talk was accepted, my first thought was “oh snap - they called my bluff.”

I believe accountability is important here - the mishandling of vulnerability info, especially “0day” vulns like this one, can result in a *lot* of weeping and gnashing of teeth. Accountability leads to responsibility, and if we take responsibility for our actions and let that influence changes in the way we do things, we will have a naturally agile and flexible way for standard operating procedures to grow and change over time. NERSC made mistakes that we’ve made moves to correct. Hopefully HPE has, too.

But perhaps more important than that is my desire to see us have a wa
moment, that you've just made an astonishing discovery: that socks m
explode. And while you were able to quickly change into sandals, you
around wearing socks, blissfully unaware of how close they are to a cat
saying "Hey buddy, notice you been exclusively in sandals lately - what
Me? Haha, I just love wearing sandals! Yeah, I know it's -1 C out, but I

I mentioned before I believe everybody's time is valuable - that's not ju
CUG, but everyone back at your home institution as well: your time is v
watched two of the smartest people I know go deep in the weeds to ge
the time everything was said and done; not to mention the dozen or so
roles! Your sysadmins already have to deal with CASTs, they shouldn't
process.

(And in HPE's defense, this is not a problem unique to them: I can't thi
report experience I've had.)



Here's the timeline of events - and as you can see, we've got some real doozies to cover. Now, you may be asking yourself…

The time from report to patch was just over one month - that's not too bad, is it Aaron?
Under normal circumstances, no - I'll get into my beef with how this went down in a few slides.

Wait - I see DVS on here - wasn't this talk about a Lustre bug?
Surprise! The same bug (ok - a different, but basically the same bug) affected DVS. I'm not gonna talk about it due to time constraints, but one thing I will say: I find it rather interesting that the Lustre vuln got a CVE, but the DVS vuln didn't. Not that I can see, anyway. Maybe we'll tug on that line of thinking later if we have time.

CVE-2023-51786

"An issue was discovered in Lustre versions 2.13.x, 2.14.x, and 2.15.x before 2.15.4, [that] allows attackers to escalate privileges and obtain sensitive information via Incorrect Access Control."

So! What's this Lustre vuln you keep going on about Aaron?

On this slide we've got the high-level description as provided by nvd.nist.gov - priv esc vulns are probably one of the bigger things on the list of "things that cause me to lose sleep", so we're already not lookin' good…

- A compromised account in conjunction with a live priv esc vulnerability is a Bad Time™
- Additionally, one needs to consider the possibility of an insider threat
  - This could be site staff, or it could be a user of your system
    - To be clear: I am not saying we should be treating our users and staff as potential threats
    - What we **should** be doing is implementing security controls that allow us to "trust but verify"
      - Ensure system logs are being sent

to your log aggregator of choice, then crc

But it's important to understand that there's a lot of nuance to vulnera
considerations for ease of exploitation and/or detection. So maybe we

CVE-2023-51786 Conditions for Exploitation

- Lustre server 2.14 - 2.15.3
- Lustre client 2.12 – 2.15.3
- Running on
  - RHEL 8.x
  - RHEL 9.x
  - SLES12 SP3 and later
  - SLES15 SPx
  - Ubuntu 18.04
  - Ubuntu 20.04
  - Ubuntu 22.04
- User namespaces enabled
  - Default since 5.10 kernel but some distros don't do this

If we pivot over to the Lustre mailing list announcement's criteria for exploitation… ah… well…

The bar ain't that high, is it. Mind you, NERSC wasn't aware at this point that the client played a role in this -

I'm glad everyone's seated, because I'm gonna go ahead and put the reproducer up on the next slide.

CVE-2023-51786 Reproducer

```
alice@login01> ls -ald /lustre/bob/
drwx------ 80 bob bob 12288 Dec  8 12:33 /lustre/bob/
alice@login01> unshare -r -U chown 0:0 /lustre/bob
alice@login01> ls -ald /lustre/bob/
drwx------ 80 alice alice 12288 Dec  8 12:33 /lustre/bob/
```


Oof.

Yeah. Let that sink in a bit.

It was really *that* simple.

Days like this you just think, “We never should have put the lightning in the rock in the first place.”

CVE-2023-51786 in a Nutshell (CVmEme?)

This is an old meme, entitled “The Internet”, but I’ll be darned if it doesn’t perfectly encapsulate the issue.

CVE-2023-51786 Root Cause

commit f05edf8e2b92e2c018aad130cd7a1dcc00a8eeee
Date: Sun Jul 12 09:15:16 2020 +0800

LU-13791 sec: enable FS capabilities
FS capabilities are not effective because they are dropped for non-root users for historical reason: they are used to be enforced before operations, but now they are checked in MDD layer only (see mdd_fix_attr()).

So: how did this happen?

The commit you see here was merged back in 2020 to support Linux capabilities on the server side.

CVE-2023-51786 Root Cause

```
old_init_ucred_common(struct mdt_thread_info *info,
/* process root_squash here. */
mdt_root_squash(info, mdt_info_req(info)->rq_peer.nid);

- /* remove fs privilege for non-root user. */
- if (uc->uc_fsuid && drop_fs_cap)
- uc->uc_cap &= ~CFS_CAP_FS_MASK;

uc->uc_valid = UCRED_OLD;
ucred_set_jobid(info, uc);
ucred_set_nid(info, uc);
```


At first glance, this looks sane - however, since the *client* wasn't updated to call setattr_prepare(), which is what's responsible for checking to see if attributes are able to be changed in a given dentry (which, for anyone like me who hadn't heard that term before: a dentry is the thing that relates inodes with file names on a system), attempts to chown items outside the username space that *should* have failed don't.

- This is an interesting situation: the bug was introduced in server code, but is patched in client code!

CVE-2023-51786 Root Cause

"The user namespace checks are performed by setattr_prepare()>chown_ok()>capable_wrt_inode_uidgid(). The names may be different for your kernel.

Unlike other filesystems, Lustre does not call this function and does not realize that the file from your test belongs to an unmapped user so chown should be prohibited."

Sure enough, HPE confirmed to us that the Lustre client is *not* calling this function and, as a result, effectively bypassing permissions checks from within user namespaces.

Vibe Check

At this point, everyone involved is feeling about like this - it's certified Bad Bad Not Good™, but (in theory) nobody else knows about it yet. I really want to emphasize "in theory": this bug existed for *three and a half years, people!* Now I'm a security dork, but security is a vast (seemingly infinite) field, and I'm no SAST/DAST guy… but it really makes you wonder how such a dead simple flaw could lurk for so long. In security, if we don't know something for a fact, we often have to make the worst-case assumption - I have to wonder if we were the first ones to discover the vuln… or simply the first to report it.

- This is a good example of something in security that makes quantifiable or objective data a bit harder to generate
- Due to the nature of the vulnerability, our ability to detect exploitation after the fact is very dependent on system configuration (not the least of which: logging policies)
    - An attacker could, in theory, do $evil, remove offending lines from relevant logs, clean up the

shell's history, then disappear

- At best, we have some indicators of compromise
- At worst, we have an undetected problem

The First Problem

Now - I'm quick to call out things I have issue with, and that includes myself. Or my employer, in this case: we made the first mistake, which was telling some other sites about the issue.

While this was quite obviously well-intentioned, without any formal information sharing agreements in place to keep groups accountable regarding further dissemination, it does run afoul of responsible disclosure.

Sound counter-intuitive? Consider this:

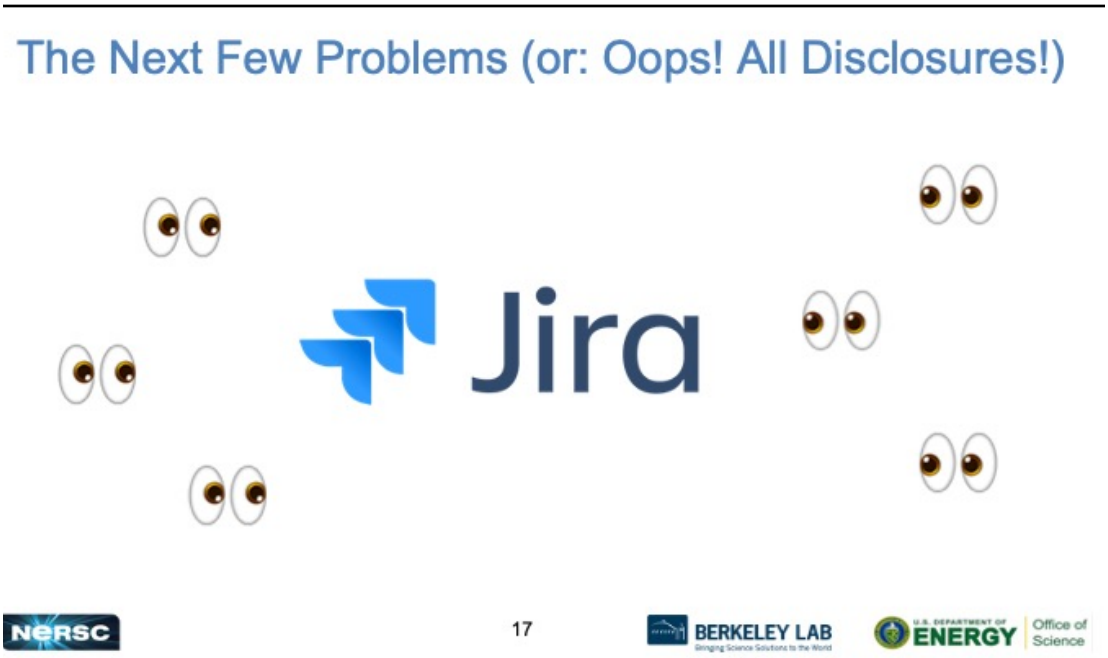
- Each person that knows about the 0day has some likelihood that they will spread the word
- As more people find out about the 0day, the likelihood of someone who would have otherwise *not* discovered this vuln increases

Trying to develop a mathematical model for that likelihood would be a really interesting exercise. :)

I said before that my role in this was in part to begrudgingly enforce th
by that. Ask anybody in this room who knows me and they will tell you
Gatekeeping information, especially security information, is something
easy task, and important, it shouldn't come at the expense of other ins

However, that's exactly what we wound up having to do - at least, to th

- While the cat was more or less out of the bag regarding the Lus
  our knowledge about the DVS vulnerability pretty well under w



But I'm getting a bit ahead of myself! You must be thinking "why are you so concerned if you only told a few other groups that, even without a formal information sharing agreement, you can *probably* trust?"

Wellllllll…

None too long after we addressed our own "internal leak", if you will, HPE said "Psh - hold my beer."

The first *real* leak of this 0day came on December 19th: HPE posted about the issue in WhamCloud's Jira…

…publicly. 😳

The post was left up long enough for Google to index it, but WhamCloud moved it into a Private issue within a few hours (rough estimate on time there.) Shouts out to Peggy over at WhamCloud, you a real one.

The Next Few Problems (or: Oops! All Disclosures!)

```
#Lustre commit 85ec44e9359b0d341e28fe7a84602e2f6b894c95
test_70() {
        local testfile=$DIR/$tdir/$tfile
        which unshare > /dev/null || skip_env "unshare is not installed"
        test_mkdir -p $DIR/$tdir || error "mkdir failed"
        touch $testfile || error "touch failed"
        chown $USER0 $testfile || error "chown failed"
        $RUNAS_CMD -u $ID1 unshare -U -r chown 0 $testfile && error "chown 0 should
have failed"
        return 0
}
run_test 70 "chown from unmapped uids should not be allowed"
```


Oh, but wait! There's more!

Describing the flaw publicly is one thing - *posting the unit test for the exploit is an entirely different pants-browning situation.*

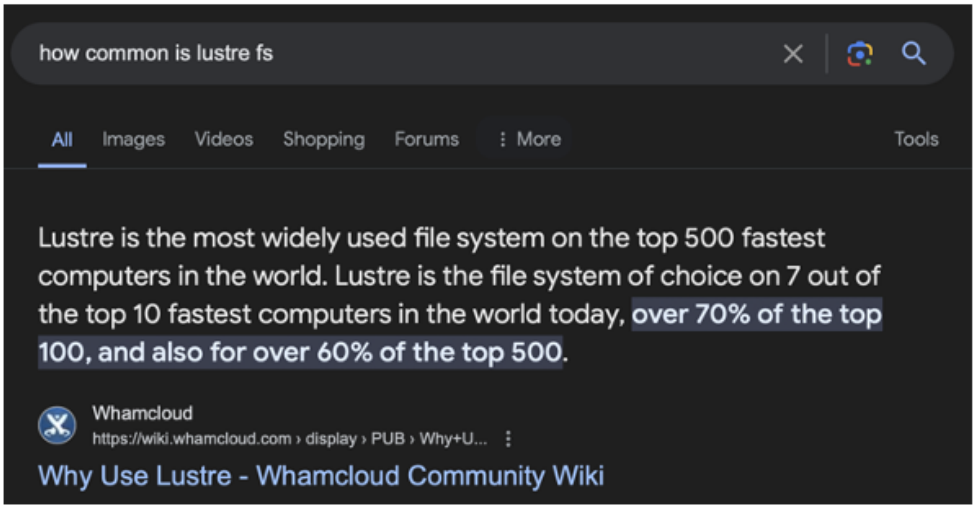
I irresponsibly didn't note how long this was posted publicly, but it doesn't really matter, and I'd like to take a moment to explain why.

Now… this may come as a surprise to some of you… but… there are *jerks* on the Internet.

- Those jerks are interested in anything from messing with you "for the lulz" to "with the intent to exploit you six ways 'til Sunday unless you meet our demands"
- Obviously, these Internet Jerks aren't tracking *every single piece of open-source code out there…*
  - **…but know this: if a codebase is popular, or serves a large portion of your industry, your code is being watched.**

I’m sure most people in the room would agree Lustre is commonly use common?

The Next Few Problems (or: Oops! All Disclosures!)



Painfully. The answer is “painfully common”.

I cannot prove what I’m about to claim, but I would stake a fair amount of whatever reputation I may have in the infosec scene on it:

The Lustre codebase is being meticulously tracked by a non-zero number of sophisticated threat groups.

- Supply chain attacks are “en vogue” at the moment
- Folks in here are likely (at least passingly) familiar with the latest high-profile supply chain attack on the xz compression utility
    - I imagine that, without the near-success of that campaign, my claim about the Lustre codebase being watched might sound significantly more… tin foil-y.

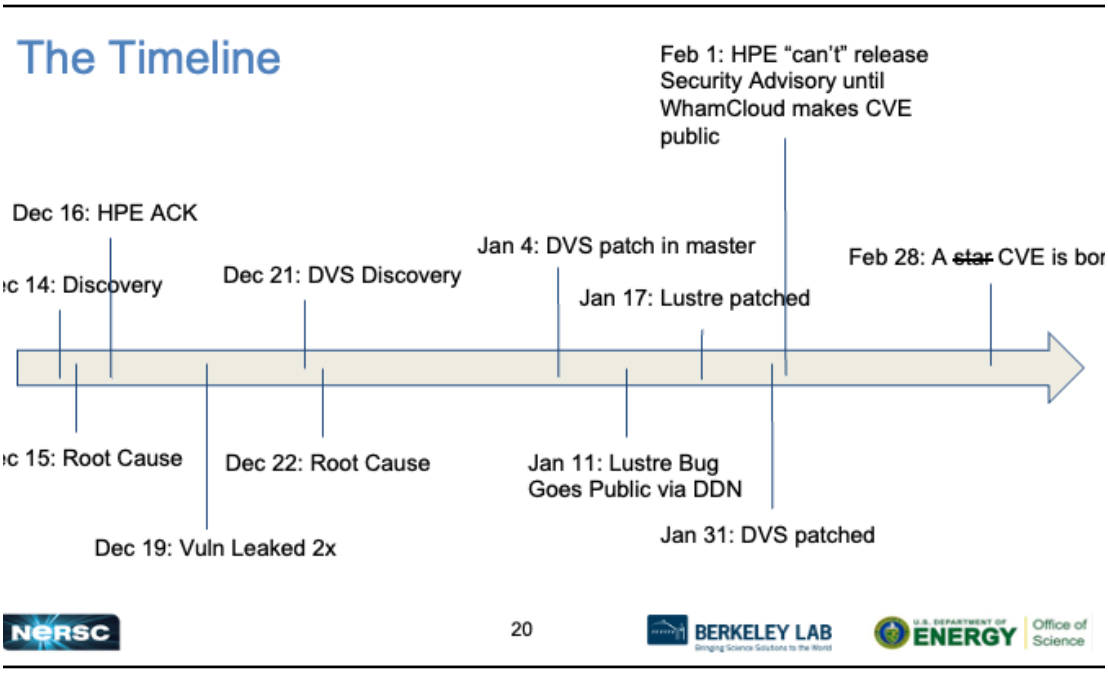
(I’m not going to say APTs, since APTs are state-sponsored actors, and that’s a *bit* further than I’d be willing to make a firm

bet on… but, for what it's worth, I think it's very likely that at least one

Why does this matter?

- It highlights a very important concept that I don't think we ofte
  (at least in part) not only by the software stacks we deploy on
  vendors are employing, the security SOP used by those vendo

To be clear, I'm not saying one should drop a vendor like a hot potato t
reliably write secure, non-trivial software with the speed and agility tha
ticket - but we need some way to hold vendors accountable.



If you think I'm overreacting, bear in mind: we're only FIVE DAYS PAST THE INITIAL DISCOVERY.

Vibe Check

The vibe is, at this time, noticeably more grim. To recap where we're at real quick:

- We have identified a 0day in one of the most common filesystems in use throughout HPC
- Exploitation only requires script-kiddie levels of skill
- NERSC (and a few other sites) are safe, but everyone else is running around carefree with their socks on…

Oh - and it's less than a week away from Christmas.

The Last Straw

- On Jan 11, DDN announces the vuln: “EXAScaler User Namespace Security Exploit (SPT-TSB-0166)”
  - Vuln is now, without any doubts, public
  - Patches are still not out (for COS, anyway - I assume they were ready upstream if DDN announced it)
  - Even *knowledge* about the vulnerability hadn’t been advertised to our satisfaction
    - No security advisory, no field notice - just a “hey, you should contact us, we have some important info for ya”
      - This would be more amicable if it were a less severe bug, or hadn’t had

knowledge of it leaked twice, or ha
public issue upstream

- On Jan 17, HPE released the fixed version of COS fo
- DVS bug is patched on Jan 31
- When asked about an update on SA issuance on Feb until Whamcloud makes the CVE public
    - Again - if this had been kept tight, I wouldn't ha
    - But it wasn't. So I do.

We finally get the public CVE notice on February 28th of thi

On Chickens and Eggs

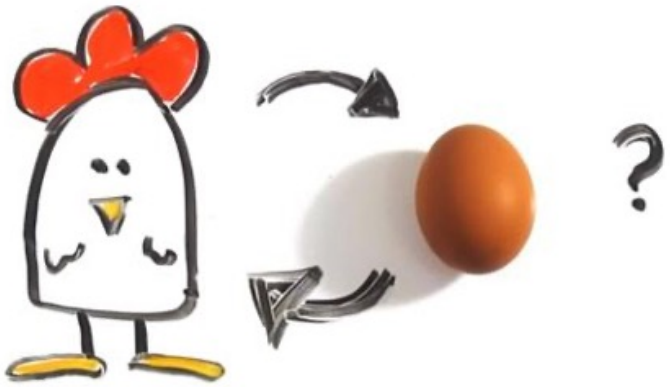
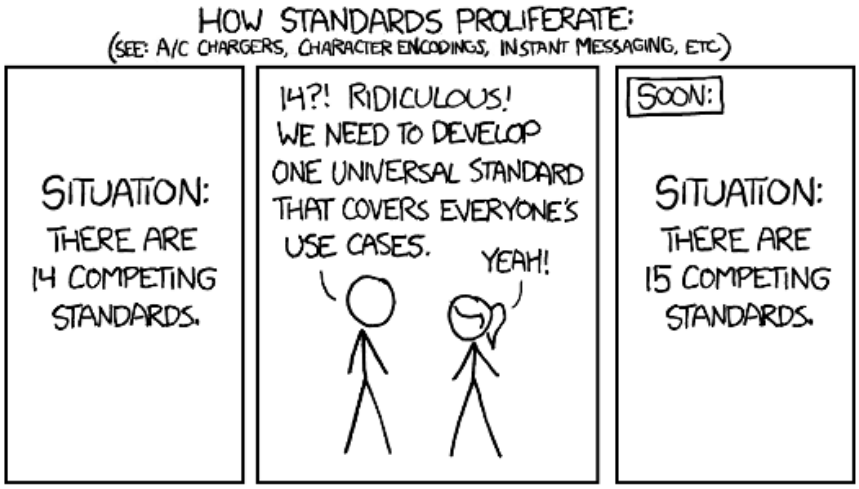


photo credit: http://www.brainpickings.org/wp-content/uploads/2013/01/chickenegg1.jpg

- Coordinated, responsible disclosure is, by and large, A Good Thing™…
  - …but doesn't provide a good mechanism for sharing intel prior to public disclosure.
- Existing reporting structure should keep stakeholders in the loop…
  - …but "too many cooks" can lead to mistakes outside of the reporter's control.
- Anyone can discover vulnerabilities…
  - …but not everyone has the bandwidth to deal with responsible disclosure.

Vulnerability Reporting is Dead - Long Live Vulnerability Reporting



At this point, I hope I've demonstrated how the current way of reporting and triaging security issues leaves something to be desired - at least in some cases.

- This is the only example *I* have direct familiarity with, but I'm willing to bet a non-zero number of you have experienced something at least tangentially similar

Now I'm using this classic XKCD comic as more a joke - I'm not suggesting we reinvent the wheel when it comes to vulnerability reporting and responsible disclosure - but I *do* think we can work together to make the current status quo *work for us*.

HPCERT- High Performance Computing Emergency Readiness Team

- Idea: a volunteer-run group of security-centric HPC professionals intake, triage and validate security concerns for HPC sites
  - Trust assurance through federation
- Once validated, HPCERT works with the vendor on behalf of the reporting institution
  - This reduces the burden on the reporting institution and, assuming a non-trivial backing of HPCSC by the HPC community, may put "more weight" on a vendor than a single site
- HPCERT will then relay relevant information to federation members
  - This absolves the reporting site of needing to constantly answer questions

I humbly submit one option: the High Performance Computing Emergency Response Team, or HPCERT.

The name is a nod to existing CERTs, or Computer Emergency Readiness Teams - in the US we have US-CERT, and here in Australia there's CERT Australia - these are government-run CERTs, but CERTs aren't necessarily always under government control: for example, AusCERT is one of the oldest CERTs in the world (founded in 1993) and is a partnership between Queensland University of Technology, Griffith University, and the University of Queensland, with the goal of creating a central source for information security and protection.

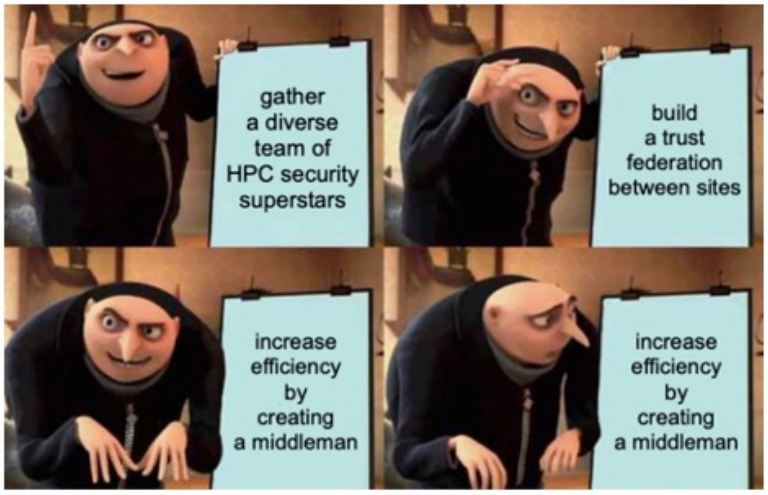
We may not all have the same security posture or risk appetite, but we all have a shared interest in the relative security of the hardware and software we leverage to push the boundaries of science. Our stacks may look different, but our goals are similar: build, support and maintain systems that get work done.

Faster than the rest of us, ideally. ;)

In security, there's something known as the CIA Triad: in essence, it po
categories of confidentiality, integrity and availability. It's that last one
you saw the reproducer earlier, your mind thought of potential confide
availability is something no less important, and isn't exactly at "the five
security isn't going to magically get us five 9s, mind you - but it sure wo
improve it.

Coming back down to earth a bit, seeing an increase in availability by fo
"pie in the sky" goal right now - to start, I'm simply looking to crowdso
security-curious!) to represent the collective interests of HPC sites whe
sites who may not have (or want to dedicate) the resources to deal wit

HPCERT- High Performance Computing Emergency Response Team



This might seem a bit counterintuitive - each new hop in a communication path brings a new potential bottleneck to rapid progress.

However, I'd posit that HPCERT's aim of not only taking the heavy lift involved with vulnerability disclosure off of individual sites' shoulders, but also (and arguably more importantly) disseminating that information in a documented, controlled fashion, could overall *improve* the experience. This would, more or less, be us catching up with the times, as it were.

Why HPCERT?

- I don't **think** this exists currently
  - REN-ISAC is a thing, but doesn't provide this service AFAIK
  - HPCSec-WG is a thing, but more focused on standards
- Getting a CVE out requires being a numbering authority
  - Vendors shouldn't be the only ones who can do this in our field
- **We are stronger and smarter when we work together**
  - How many of you think your security team is sufficiently staffed? :)
  - Many diverse minds weighing in on a problem that needs to be solved generally yields a better solution

There are a few reasons why I think we should do this, but to be honest it's not completely clear to me whether or not it would work. CERTs are nothing new, so at face value it seems like it might be worth pursuing, but I'm still a bit too green in the HPC world to know for certain, let alone exactly how we might run it.

What is Needed for HPCERT Work?

YOU

(well - the royal "you" - or the actual "you", if you're security-oriented!)

(ok - we should probably have a plan, too)

People, firstly.

Time, secondly - Rome wasn't built in a day and all that. Furthermore, while this idea *sounds* pretty neat, it might suck *worse* than the current way of doing things - but in theory, if we:

- Form a CUG SIG
  - Endgame I see this as a group that would work with *any* vendor on *any* vuln disclosure, but starting it at CUG lets us start small and use HPE as a starting point for figuring out how this might work
  - Furthermore, I think we can all agree that, even if HPCERT isn't a good idea after all, a SIG that covers HPC security concerns and issues is of great value to this community

- and can be a launching pad for additional efforts
- Take some time to model what HPCERT might look like
  - Volunteer structure, infrastructure, purpose, etc
- Information sharing / general trust agreement
- Get buy-in from interested sites
  - Volunteer time chief among desired support
    - Will need to ensure we have a proper ratio of volunteers:federation size
- Get buy-in from vendors
  - Resources for validating reports chief ask
    - Volunteers may not be able to reproduce without proper licensing
- …and all the other fun things that come with operating a v
  - I just think this is a good idea - I've thus far made a living somehow fooling p ground.

…then we may very well have something in a year or two!

Conclusions

- The current state of the responsible disclosure process has room for improvement
  - Not specifically in HPC - this is generally true for any vertical
- In a perfect world, every research site would have adequate staffing for security
  - In the real world, a site's needs for in-house security knowledge may fluctuate, and keeping adequate staff for "peak dumpster fire times" may not be feasible
- Security sucks, and can be hard to implement when it's at odds with the speed of progress, but it is **everybody's** responsibility
  - Whether or not HPCERT becomes "a thing", having a SIG dedicated to security almost assuredly in everybody's best interest

So… what are our takeaways today?

First, at least to some degree: we have to admit that there are some issues with responsible disclosure. Even well-intended actions can result in consequences, and not all the variables in this equation are in control of the reporting entity.

- Defense in depth is a common security paradigm since there's no silver bullet for solving security problems, and protections need to be layered such that the failure of one control doesn't result in a loss of security

Second, that security needs are elastic - and subject to EXTREME shifts in need at that!

- In industry, this need is generally covered by having an incident response firm on retainer
- Neither HPCERT nor a SIG are going to replace the need for that - however, it may reduce the likelihood of you needing to activate said firm

Lastly, that security sucks - *especially* in the world of HPC - and as a sec

- In industry, cycles not being spent on whatever the org does ge
  - That translation works in HPC as well, but perhaps more *science*
  - Nobody wants performance regressions - but by the san risk appetite
    - While HPCERT would be more of an incident resp come together to outline the issues each site is h (and brains!) on the problem than if we were to
    - I know I'm preaching to the choir here, but we re together

Future Work

1. Form CUG SIG
1. Work with others to determine need, purpose for HPCERT
1. ???
1. (non)profit

Thank you!

Questions? Comments? Daps? High-fives?

ascantlin@lbl.gov

Appendix

- Lustre CVE: https://nvd.nist.gov/vuln/detail/CVE-2023-51786
- (Lack of) DVS CVE: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=DVS
- Lustre mailing list announcement: http://lists.lustre.org/pipermail/lustre-announce-lustre.org/2024/000270.html
- Breaking Lustre commit: https://review.whamcloud.com/c/fs/lustre-release/+/39399
- Fixing Lustre commit: https://review.whamcloud.com/c/fs/lustre-release/+/53503/2/lustre/llite/llite_internal.h
- HPC Security Working Group: https://csrc.nist.gov/projects/high-performance-computing-security