

# Secure Erase with hdparm & Raid Alignment

Alex Owen

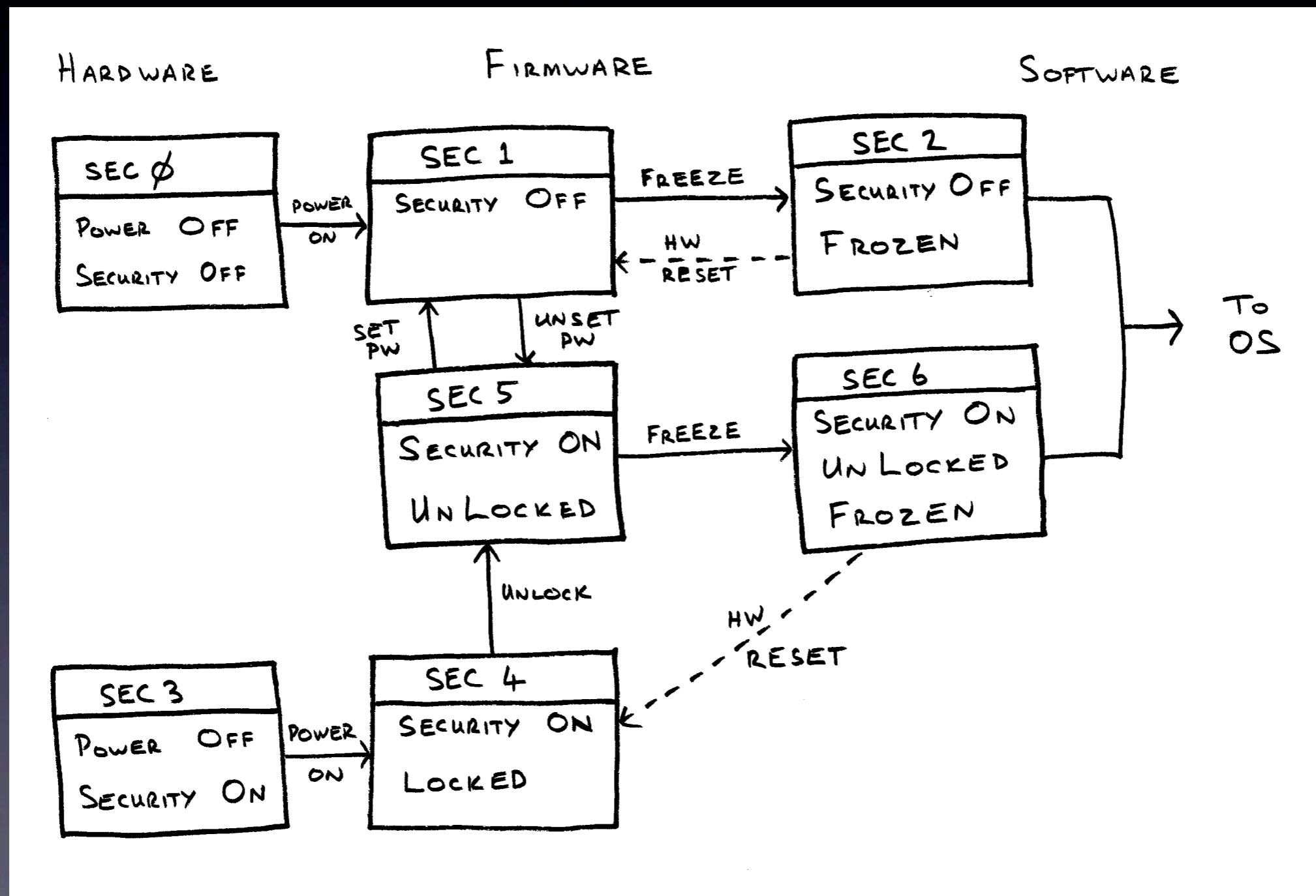
Particle Physics Research Center  
Queen Mary University of London

[r.a.owen@qmul.ac.uk](mailto:r.a.owen@qmul.ac.uk)

# Previously at HepSysman...

- In 2014 I was bricking drives trying to use hdparm to run secure erase

# ATA Security: Target SEC5



# Getting to SEC5

- SATA Disk
- External SATA enclosure (eSATA)
- NOT USB enclosure => bricked disk
- Disk => Enclosure after boot => SEC1
- hdparm => SEC5

# hdparm -I \$DEV

## Security:

Master password revision code = 65534

supported

not enabled

not locked

frozen

not expired: security count

supported: enhanced erase

40min for SECURITY ERASE UNIT.

40min for ENHANCED SECURITY ERASE UNIT.

# hdparm key commands

DEV=/dev/sdb

#show security/locked/frozen state

hdparm -l \$DEV

#set password "ERASE" /Enable Security

hdparm --user-master u --security-set-pass ERASE \$DEV

#do standard erase using password

time hdparm --user-master u --security-erase ERASE \$DEV

#if erase fails (check with hdparm -l) then unset password

hdparm --user-master u --security-disable ERASE \$DEV

# hdparm Bugs!

- Use hdparm-9.45 or above
  - Probably not in current distros
  - Easy compile
- There was a 2 hour max timeout in previous versions. Fix was implemented in parts over a couple of versions.

# Script

```
#!/bin/bash
set -e
#global vars
DEV=$1
#HDPARM=/sbin/hdparm
HDPARM=/var/tmp/hdparm/hdparm-9.45/hdparm
#functions
usage(){
echo
echo Usage: $(basename $0) /dev/XXX
echo
}
isSecurityDisabled(){
$HDPARM -I $1 |sed -n -e '/Security:/,/enabled/s/enabled/enabled/p'|grep -q not
}
error(){
message="$1"
exitcode="$2"
echo "ERROR: $message"
echo $(date) END: $0 $1 - EXITCODE: $exitcode
exit $exitcode
}
# main
if [ $(id -u) -ne 0 ] ; then
echo MUST RUN AS ROOT
exit 2
fi
if [ $# -ne 1 ] ; then
usage
exit 1
fi
echo $(date) START: $0 $1
smartctl -o off -S off -s off $DEV
if ! isSecurityDisabled $DEV; then
error "Disk is already password protected" 1
fi
$HDPARM --user-master u --security-set-pass ERASE $DEV
sleep 1
if isSecurityDisabled $DEV; then
error "Could not prepare disk for secure erase" 2
fi
ES=0;wipe=0
time $HDPARM --user-master u --security-erase ERASE $DEV || ES=$?
echo Exit Status: $ES
sleep 1
if [ $ES != 0 ] ; then wipe=1 ; fi
if ! isSecurityDisabled $DEV ; then
wipe=1
$HDPARM --user-master u --security-disable ERASE $DEV || error "Cannot disable security" 3
fi
smartctl -s on -S on -o on $DEV
if [ $wipe != 0 ] ; then
echo dd if=/dev/zero of=$DEV bs=1M
time dd if=/dev/zero of=$DEV bs=1M || /bin/true #as gives error at end of device
fi
$HDPARM -I $DEV
sleep 1
WAITMIN=$(smartctl -c $DEV | sed -n -e '/Extended/,/polling time:/s%.*( *\[0123456789*\]).*%\1%p')
smartctl -t long $DEV
[ -z $WAITMIN ] && WAITMIN=60
sleep $(( $WAITMIN * 62)) # allow 62 seconds per minute to allow a margin for error
smartctl -a $DEV || /bin/true
smartctl -H $DEV || /bin/true
echo $(date) END: $0 $1
```



# Enhanced Erase

- Also erases re-mapped sectors

```
hdparm --user-master u --security-erase-enhanced ERASE $DEV
```

# Thanks for listening!

Questions?

# Raid Alignment

[https://pprc.qmul.ac.uk/~owen/raid\\_align.py](https://pprc.qmul.ac.uk/~owen/raid_align.py)

# --help

```
$ raid_align.py --help
```

```
raidalign
```

```
-c --raidchunk=64KiB
```

```
-d --disk=12
```

```
-r --raid=6
```

```
-p --pe=4MiB (LVM Physical Extents)
```

```
-s --size=1TiB (required LV size)
```

```
-b --block=4096B (ext2 block size)
```

```
-o --output=[all|lvcreate|lvextend|ext3|ext4|xfs] (default is 'all'  
implies -q)
```

```
-h --help (usage)
```

```
-q --quiet (Don't prompt for missing input)
```

# Interactive Run

```
$ raid_align.py
raidchunk [64KiB]:
disk [12]:
raid [6]:
pe [4MiB]:
size [1TiB]:
block [4KiB]:
  raid 6      raidchunk 65536
  disk 12    datadisk 10
  pe 4194304  size 1099511627776
  block 4096
You asked for: 1 TiB = 1099511627776 B
Which rounded up for coincidence is: 1 TiB = 1099515822080 B
Which is 262145 pe's and 1677728 raid stripe_widths
```

Example command fragments:

```
pvcreate --dataalignment 655360B
vgcreate --dataalignment 655360B
lvcreate -L 1099515822080B
mkfs.ext3 -b 4096 -E stride=16,stripe-width=160
mkfs.xfs -d sunit=128,swidth=1280 -l sunit=128
```

If extending an LV by the requested size:  
lvextend -L +1099515822080B