ZONE MINDER

Using ZoneMinder, Debian Linux, and BackBlaze to solve video monitoring problems

Linux Video Security

Project Background

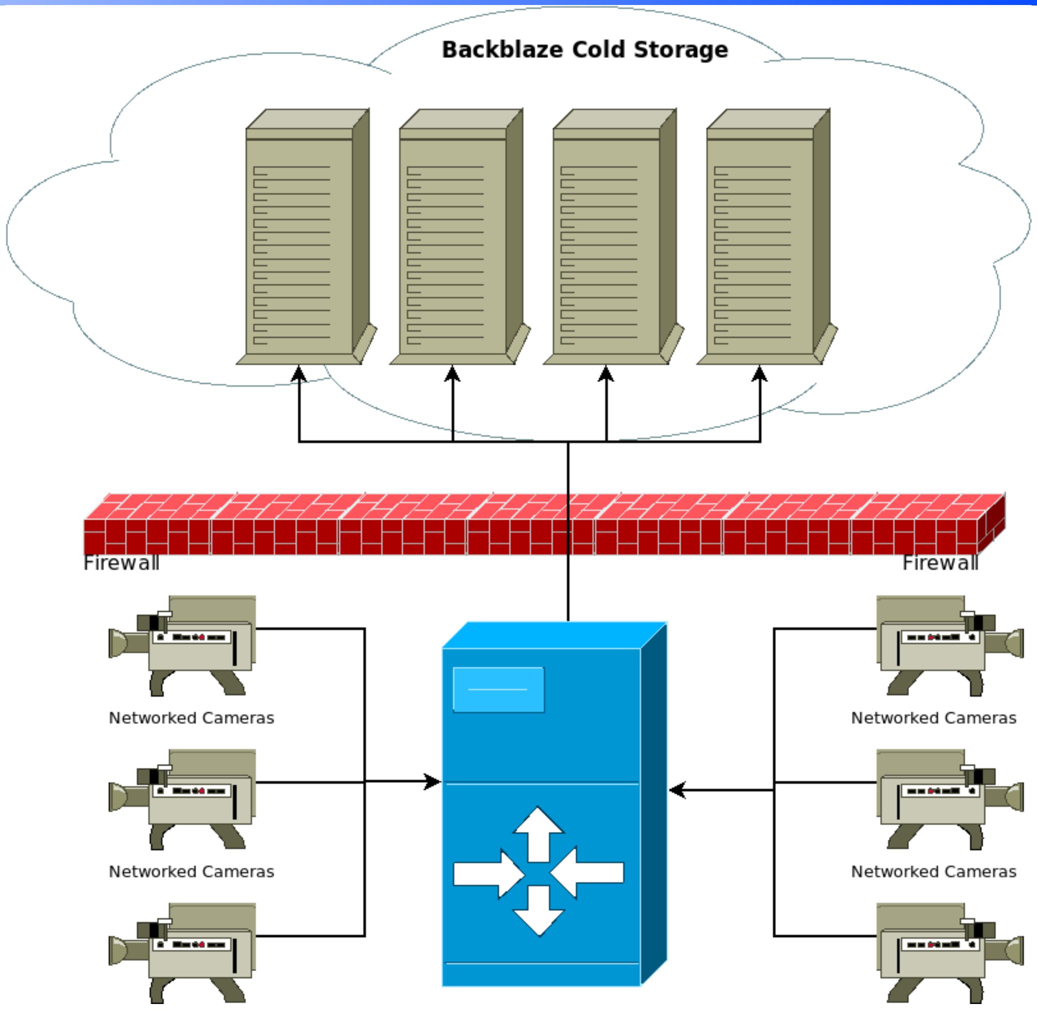
- Startup Environment
  - Single devops/sysadmin
  - Low budget
- Security Monitoring Needed
- Risk of liability without record of events

Project Parameters

- Must be Scalable
  - Deployment to other locations
  - Retain video indefinitely
- Must be Accessible
  - Retention of video for legal/liability purposes
- Must be Secure
  - Electronic attack mitigation
  - Physical attack mitigation
- Must be Automated
  - Set up, document, and ignore

Architecture

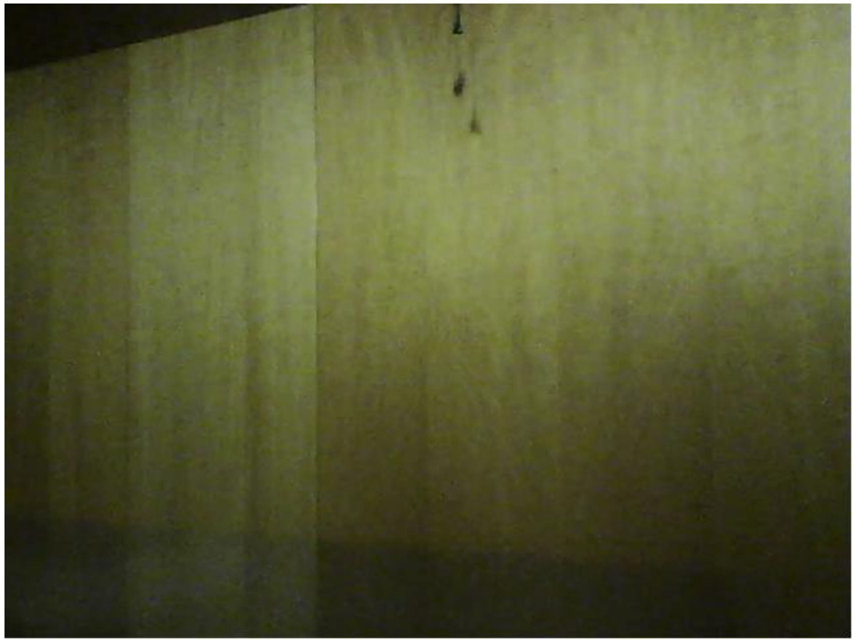
- IP Cameras
  - DCS-934-L
  - DCS-932-L
- ZoneMinder Server
  - Debian 8
- Backblaze
  - B2 Cloud Storage



Hardware Hack – DCS-93x

- Visual Artifacts in Low Light
  - Fix by with a 470µF capacitor across C38 and L8

Before


After

From http://forums.dlink.com/index.php?topic=52839.0

Generic Server Setup

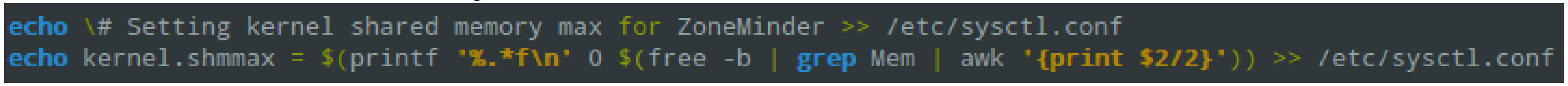
- Install and tune Debian 8
  - Create SSH user
    - Set RSA Pubkey auth only
  - Disable root SSH
  - Set system timezone
  - Remove systemd
  - Configure update autoinstallation
    - Update and reboot server weekly

Security

- Iptables
- Fail2ban
  - Monitor Apache
  - Monitor SSH
  - Monitor sudo
- SSH
  - IP whitelist
  - RSA Pubkey auth only – no passwords
- Read-only .ssh directory
- Port forwarding

Install ZoneMinder

- Add jessie-backports to /etc/apt/sources.list
  - Import GPG keys
  - Pin backports package priority
- Set shared memory maximum



```
echo \# Setting kernel shared memory max for ZoneMinder >> /etc/sysctl.conf
echo kernel.shmmax = $(printf '%.*f\n' 0 $(free -b | grep Mem | awk '{print $2/2}')) >> /etc/sysctl.conf
```

- Install prerequisite packages
  - apache2, php5, pear, mariadb
- Install ZoneMinder
  - Import database
  - Enable Apache2 modules

Apache2 Config

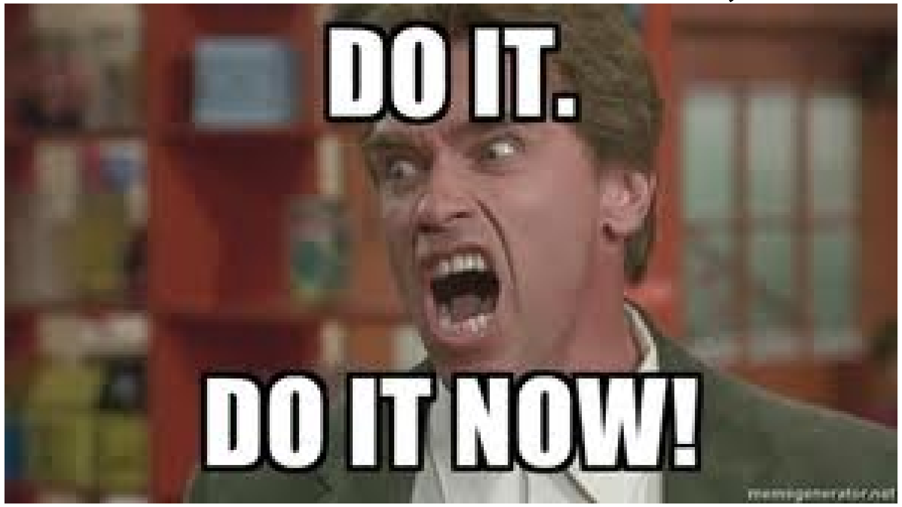
- Proxy, LetsEncrypt certificate, HTTPS only

```
<VirtualHost *:443>
  ServerName redthreadstudios.org
  ServerAlias zm.redthreadstudios.org

  ScriptAlias /cgi-bin "/usr/lib/zoneminder/cgi-bin"

  <Directory "/usr/lib/zoneminder/cgi-bin">
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    AllowOverride All
    Require all granted
  </Directory>

  DocumentRoot /usr/share/zoneminder/www
  <Directory /usr/share/zoneminder/www>
    php_flag register_globals off
    Options Indexes FollowSymLinks
    <IfModule mod_dir.c>
      DirectoryIndex index.php
    </IfModule>
  </Directory>
  SSLEngine On

  SSLProtocol all -SSLv2 -SSLv3
  SSLHonorCipherOrder on
  SSLCipherSuite EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA384:EECDH+ECDSA+SHA256:EECDH+aRSA+SHA384:EECDH+aRSA
+SHA256:EECDH+aRSA+RC4:EECDH:EDH+aRSA:RC4:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!PSK:!SRP:!DSS:!RC4

  SSLCertificateFile /etc/letsencrypt/live/zm.redthreadstudios.org/cert.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/zm.redthreadstudios.org/privkey.pem
  SSLCertificateChainFile /etc/letsencrypt/live/zm.redthreadstudios.org/chain.pem
</VirtualHost>
```

```
<VirtualHost *:80>
    ServerName  redthreadstudios.org
    ServerAlias zm.redthreadstudios.org octopi.redthreadstudios.org
    ServerAdmin webmaster@redthreadstudios.org

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}

</VirtualHost>

<VirtualHost *:80>
    ServerAlias *
    ServerAdmin webmaster@redthreadstudios.org

    <Location />
        Order deny,allow
        Deny from all
    </Location>
</VirtualHost>
```

LetsEncrypt

- HTTPS is the only way
- Always use HTTPS
- There's no excuse to not HTTPS everything
- Seriously, certificates are free, use HTTPS



Camera Configuration

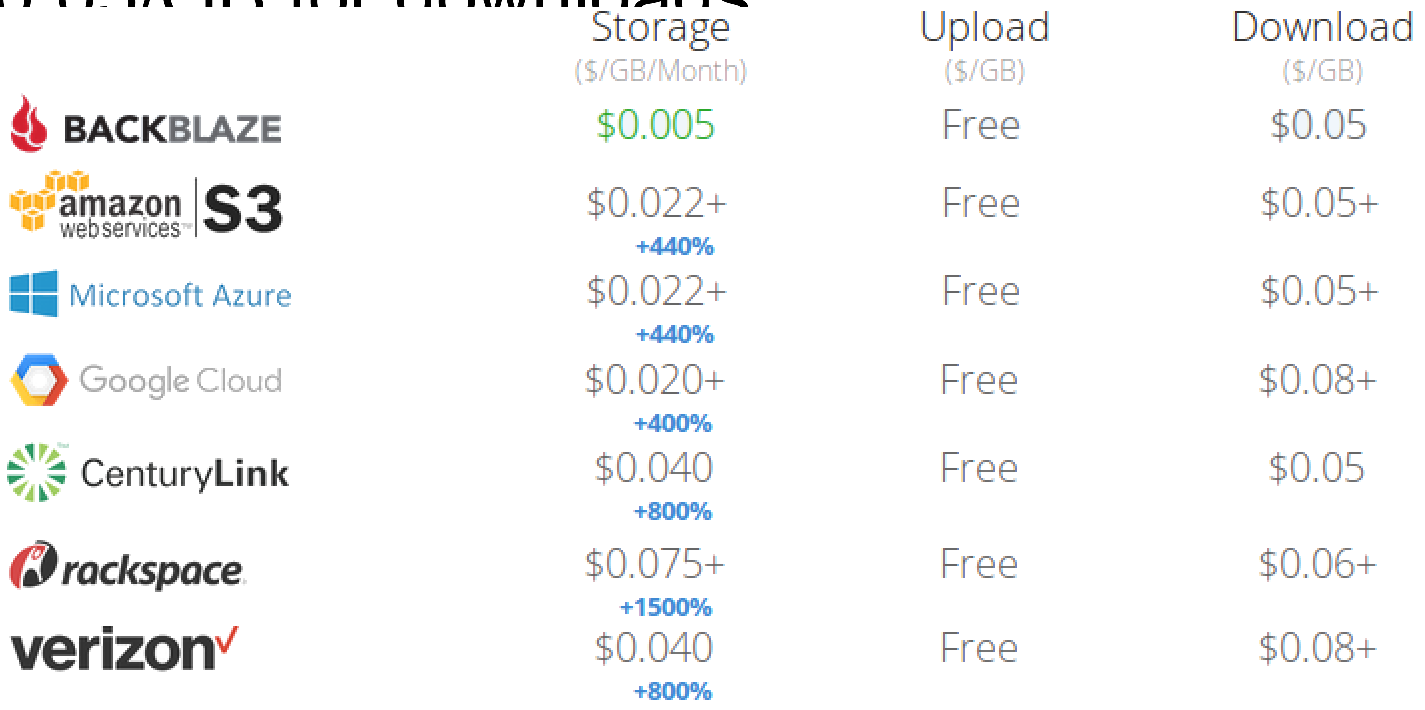
- Set output format
- Configure security
  - Disable unneeded options (eg builtin FTP)
  - Require authentication
    - Use “user:password@ip.address” in ZoneMinder
- Set night mode always on

ZoneMinder Configuration

- Scheduled recording with run states
  - Uses zmpkg.pl and cron
  - Motion detection vs run states
- Set up monitor groups
- Filters and background execution

Backblaze B2 Cloud Storage

- Low cost long term storage
  - $0.005/month per GB stored
  - $0.05/GB for downloads

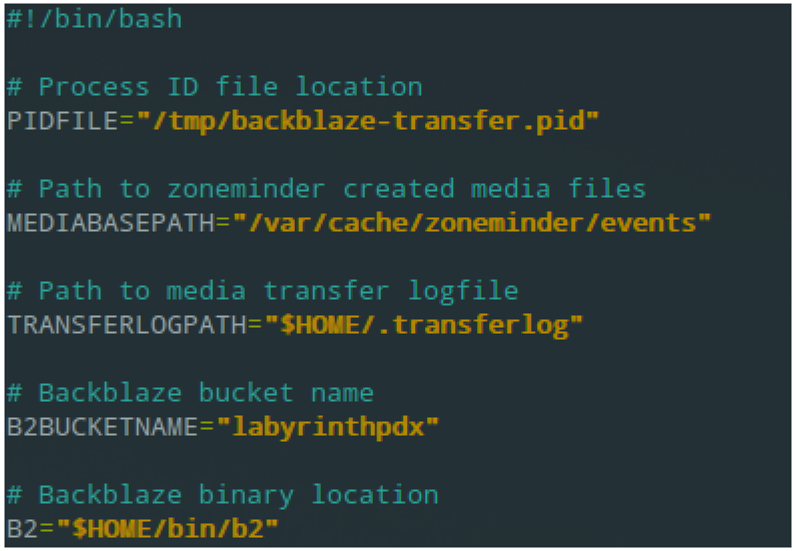


	Storage ($/GB/Month)	Upload ($/GB)	Download ($/GB)
BACKBLAZE	$0.005	Free	$0.05
amazon web services S3	$0.022+ +440%	Free	$0.05+
Microsoft Azure	$0.022+ +440%	Free	$0.05+
Google Cloud	$0.020+ +400%	Free	$0.08+
CenturyLink	$0.040 +800%	Free	$0.05
rackspace	$0.075+ +1500%	Free	$0.06+
verizon	$0.040 +800%	Free	$0.08+

Numbers from https://www.backblaze.com/b2/cloud-storage-providers.html

Backblaze CLI Automation

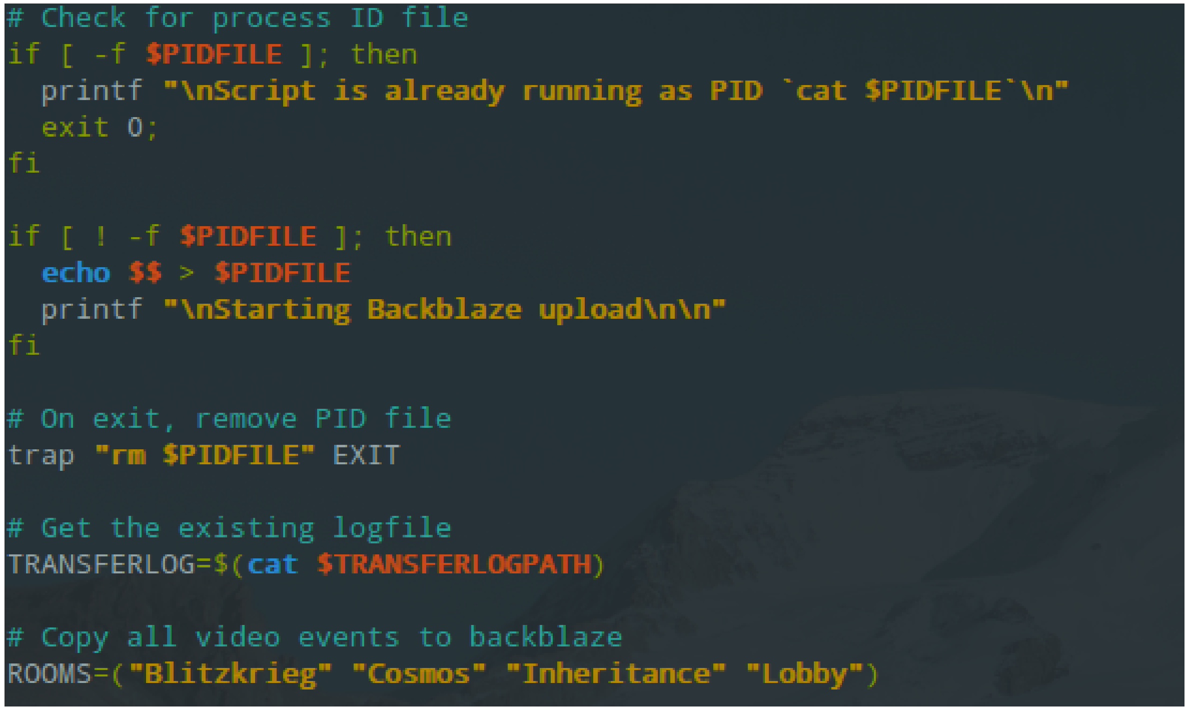
- Set up variables
  - Process ID file
  - Location of video
  - Logfile location
  - Backblaze bucket name
  - Backblaze binary location



```
#!/bin/bash

# Process ID file location
PIDFILE="/tmp/backblaze-transfer.pid"

# Path to zoneminder created media files
MEDIABASEPATH="/var/cache/zoneminder/events"

# Path to media transfer logfile
TRANSFERLOGPATH="$HOME/.transferlog"

# Backblaze bucket name
B2BUCKETNAME="labyrinthpdx"

# Backblaze binary location
B2="$HOME/bin/b2"
```

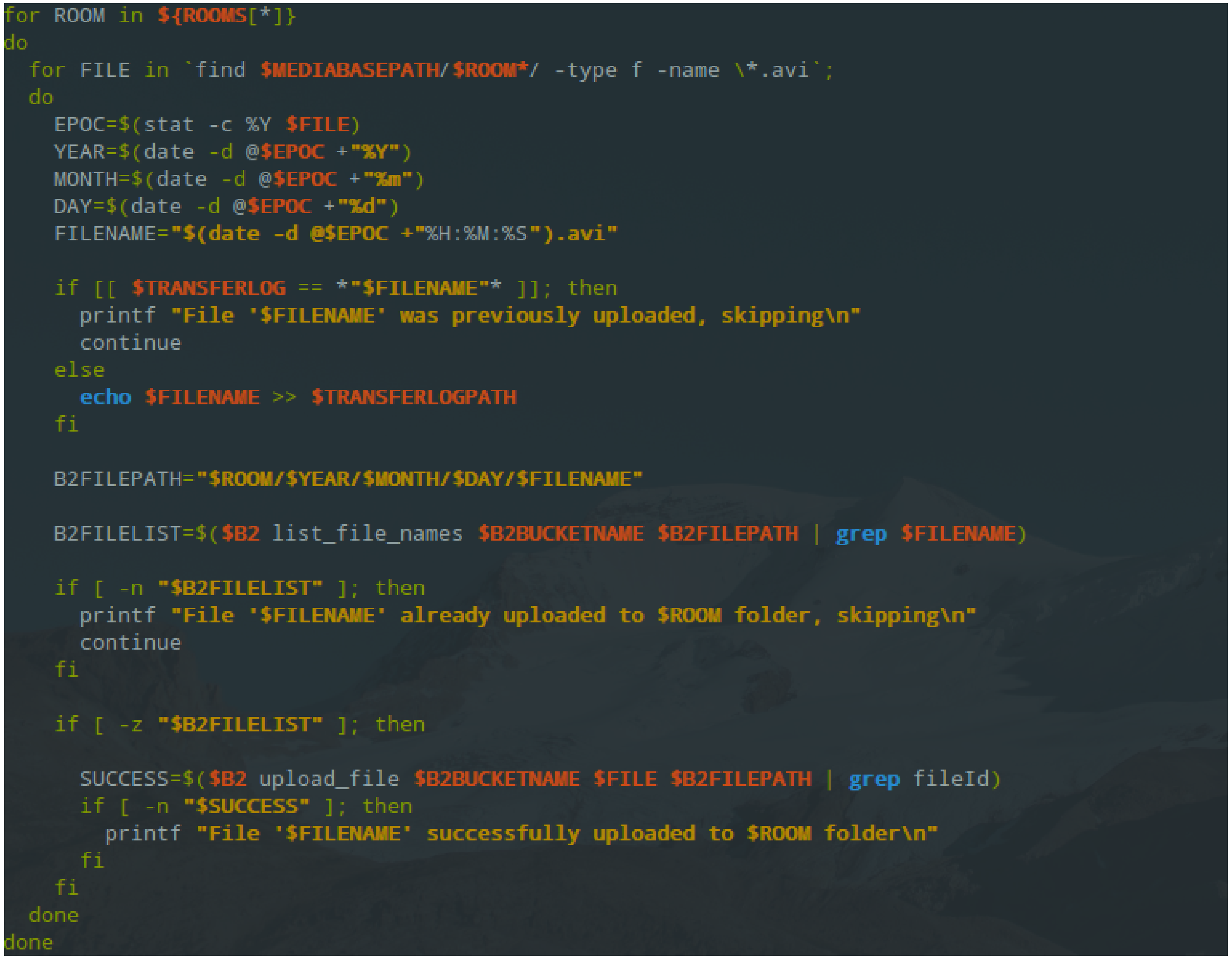
Backblaze CLI Automation

- Eliminate double running
  - Use a PID file
  - Use bash exit trapping



```
# Check for process ID file
if [ -f $PIDFILE ]; then
  printf "\nScript is already running as PID `cat $PIDFILE`\n"
  exit 0;
fi

if [ ! -f $PIDFILE ]; then
  echo $$ > $PIDFILE
  printf "\nStarting Backblaze upload\n\n"
fi

# On exit, remove PID file
trap "rm $PIDFILE" EXIT

# Get the existing logfile
TRANSFERLOG=$(cat $TRANSFERLOGPATH)

# Copy all video events to backblaze
ROOMS=("Blitzkrieg" "Cosmos" "Inheritance" "Lobby")
```

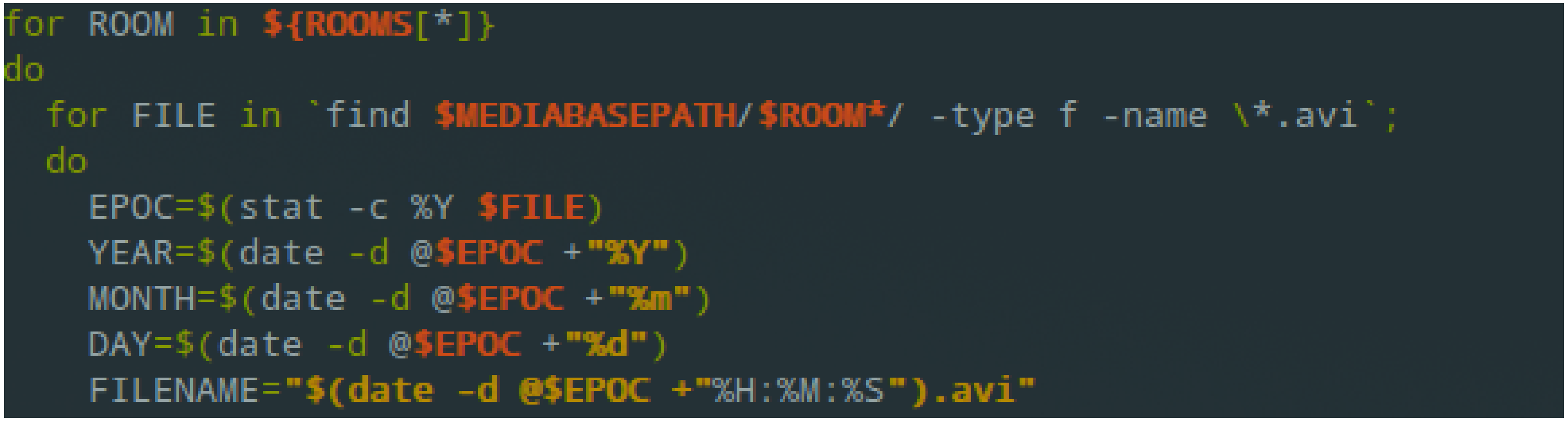
Backblaze CLI Automation



```
for ROOM in ${ROOMS[*]}
do
  for FILE in `find $MEDIABASEPATH/$ROOM*/ -type f -name \*.avi`;
  do
    EPOC=$(stat -c %Y $FILE)
    YEAR=$(date -d @$EPOC +"%Y")
    MONTH=$(date -d @$EPOC +"%m")
    DAY=$(date -d @$EPOC +"%d")
    FILENAME="$(date -d @$EPOC +"%H:%M:%S").avi"

    if [[ $TRANSFERLOG == *"$FILENAME"* ]]; then
      printf "File '$FILENAME' was previously uploaded, skipping\n"
      continue
    else
      echo $FILENAME >> $TRANSFERLOGPATH
    fi

    B2FILEPATH="$ROOM/$YEAR/$MONTH/$DAY/$FILENAME"

    B2FILELIST=$($B2 list_file_names $B2BUCKETNAME $B2FILEPATH | grep $FILENAME)

    if [ -n "$B2FILELIST" ]; then
      printf "File '$FILENAME' already uploaded to $ROOM folder, skipping\n"
      continue
    fi

    if [ -z "$B2FILELIST" ]; then

      SUCCESS=$($B2 upload_file $B2BUCKETNAME $FILE $B2FILEPATH | grep fileId)
      if [ -n "$SUCCESS" ]; then
        printf "File '$FILENAME' successfully uploaded to $ROOM folder\n"
      fi
    fi
  done
done
```

Backblaze CLI Automation

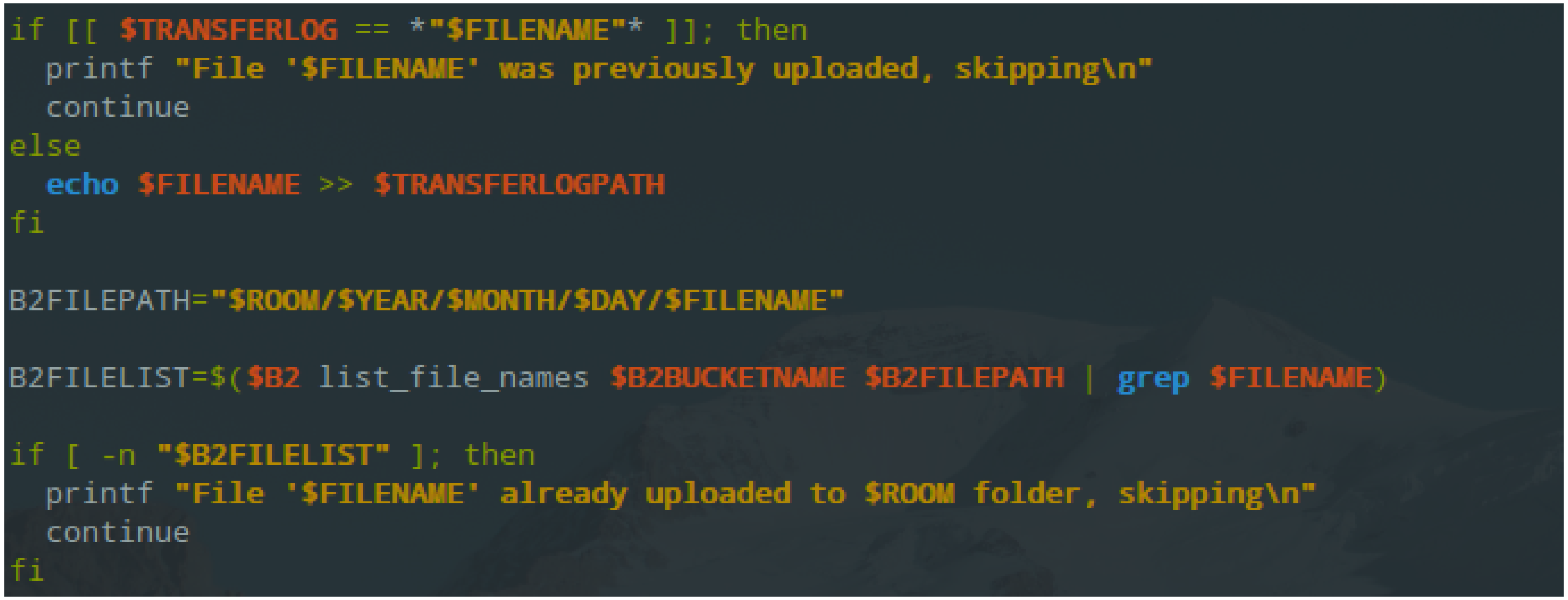
- Iterate through rooms
  - Locate all .avi files
  - Build filename based on video modification date



```
for ROOM in ${ROOMS[*]}
do
  for FILE in `find $MEDIABASEPATH/$ROOM*/ -type f -name \*.avi`;
  do
    EPOC=$(stat -c %Y $FILE)
    YEAR=$(date -d @$EPOC +"%Y")
    MONTH=$(date -d @$EPOC +"%m")
    DAY=$(date -d @$EPOC +"%d")
    FILENAME="$(date -d @$EPOC +"%H:%M:%S").avi"
```

Backblaze CLI Automation

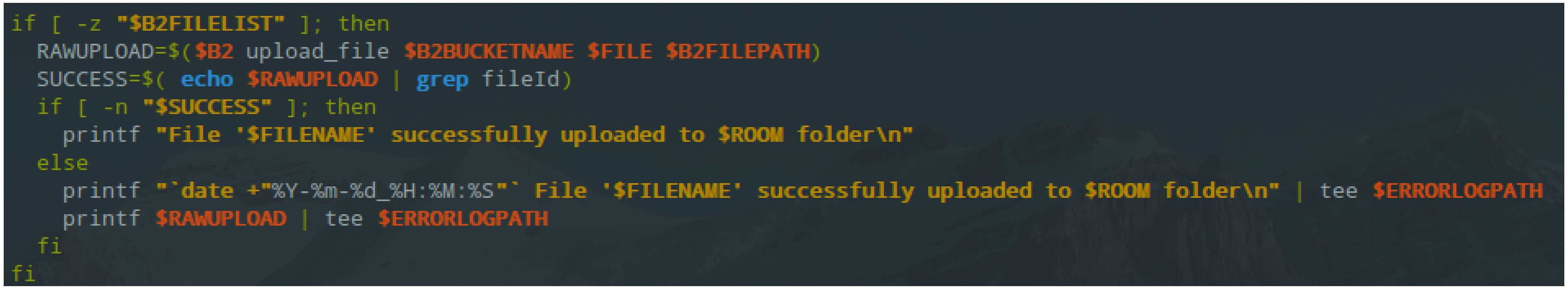
- Double verify before uploading
  - Check local logfile first
  - Query Backblaze second



```
if [[ $TRANSFERLOG == *"$FILENAME"* ]]; then
  printf "File '$FILENAME' was previously uploaded, skipping\n"
  continue
else
  echo $FILENAME >> $TRANSFERLOGPATH
fi

B2FILEPATH="$ROOM/$YEAR/$MONTH/$DAY/$FILENAME"

B2FILELIST=$($B2 list_file_names $B2BUCKETNAME $B2FILEPATH | grep $FILENAME)

if [ -n "$B2FILELIST" ]; then
  printf "File '$FILENAME' already uploaded to $ROOM folder, skipping\n"
  continue
fi
```

Backblaze CLI Automation

- Upload and verify
  - Log upload errors



```
if [ -z "$B2FILELIST" ]; then
  RAWUPLOAD=$($B2 upload_file $B2BUCKETNAME $FILE $B2FILEPATH)
  SUCCESS=$( echo $RAWUPLOAD | grep fileId)
  if [ -n "$SUCCESS" ]; then
    printf "File '$FILENAME' successfully uploaded to $ROOM folder\n"
  else
    printf "`date +"%Y-%m-%d_%H:%M:%S"` File '$FILENAME' successfully uploaded to $ROOM folder\n" | tee $ERRORLOGPATH
    printf $RAWUPLOAD | tee $ERRORLOGPATH
  fi
fi
```

Backblaze CLI Automation

- Future improvements
  - Better logging
    - File Ids
    - Upload times
  - Log rotation
  - Video merging for clustered events
  - Recording schedule based on calendar

Future Additions

- Physical Security Features
  - Locking server cabinet
  - Intruder alarm
- Electronic Security Features
  - Two factor authentication
  - Hard Drive Encryption
  - Intermediary upload server
  - Disable destructive commands
  - SELinux permissions

Q&A

- Questions?
- Comments?
- Random Rhyming Remarks?