Recovering a Failed QNAP Raid Volume

How to recover data from QNAP drives using testdisk from SystemRescueCd

Pre-flight

Given the following scenario:

QNAP server was factory reset, clearing the software RAID information on the QNAP OS.
As such, all drives in the RAID were essentially orphaned.
Data on the drives remained intact.

Recovery Options:

In order to recover the information, we could proceed via many troubleshooting pathways, two of which I list below:

– Rebuilding the software RAID
– Recovering the data directly from the drives

I chose the second option, since I wasn't too handy with administration of the Linux Multiple Device Driver (**MD**), aka software RAID.
In this article, we will be recovering the data from **ONE** drive at a time, so it is best to plug in **ONLY ONE** of drives to be recovered, along with a **spare** drive on which the recovered data will be copied to.

Recovery Software:
We will be using **SystemRescueCD** to perform the data recovery

I assume the following:
You've already booted the SystemRescueCD

You either have console or ssh access (or whatever other means) to the SystemRescueCD shell
You have the drive to be recovered and a spare plugged in to your system

Lastly, this is key in **Understanding QNAP volumes:**
QNAP utilizes Logical Volume Management (**LVM**) and the Linux **MD** software RAID technologies to manage its storage devices.
Partition 3 Holds all the **data** on any given drive
Keep this in mind as you start digging for your data on the QNAP drives.

Identify the Destination Drive

Before going through the recovery, you must prep the directory on which you will be copying the recovered data to.
With the specs on your hard drive already in mind, issue the list hardware command (**lshw**) to determine the device name to the drive:
lshw -short -c disk
Once you match the device information to that of the spare drive, you can proceed to initialize (wipe/clean) the drive or mount it if it's already prepared.

If the drive is already initialized, skip the next step, otherwise proceed …

Prepare the Destination Drive

You can initialize the drive for use on the SystemRescueCD as follows:
fdisk <device_name>, e.g. **fdisk /dev/sda**
Follow the prompts to create a **Linux Partition**
Note: Once the partition is created, the device you'll actually be acting against is <device_namelogical_partition_number>, e.g. **/dev/sda1**
Once you've written the changes to the disk, you can proceed

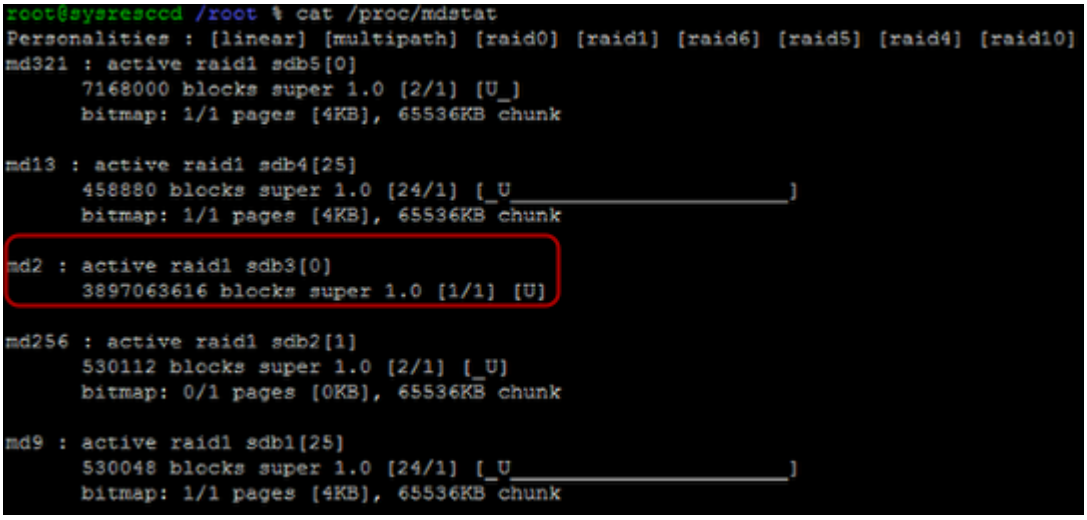
to create the filesystem on the drive:
mkfs -t <fs_type> <device_namelogical_partition_number>, e.g.
mkfs -t ext4 /dev/sda1
or
mkfs.<fstype> <device_namelogical_partition_number>, e.g.
mkfs.ext4 ext4 /dev/sda1
Once the filesystem has been created, you can mount it.
Do so first by creating a directory on which the drive will be mounted, e.g.:
mkdir /mnt/recovery

Mount the Destination Drive

Mounting the drive is quite easy, simply invoke the **mount** command, e.g.:
mount -t ext4 /dev/sda1 /mnt/recovery

Your destination drive is now ready to be used!

Identify the Data Partition on the Source Drive



The following commands are to be issued from the SystemRescueCD session:

First, we need to determine what MD volumes the SystemRescueCD

has detected.
You can do so by displaying the contents of the mdstat file under /proc as follows:

```
cat /proc/mdstat
```

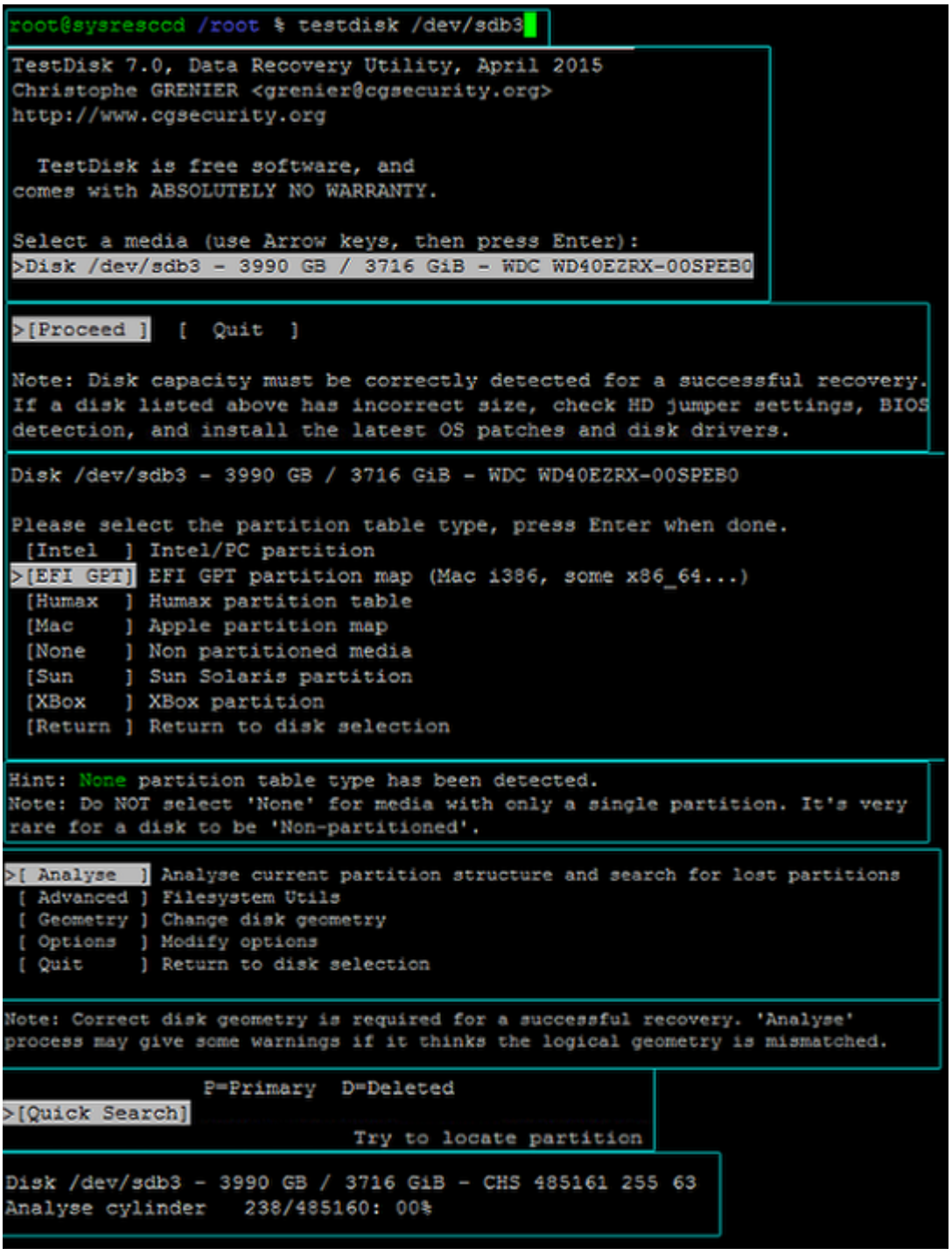
Samlpe Output:

```
        Personalities : [linear] [multipath] [raid0] [raid1]
[raid6] [raid5] [raid4] [raid10]
       md321 : active raid1 sdb5[0]
             7168000 blocks super 1.0 [2/1] [U_]
             bitmap: 1/1 pages [4KB], 65536KB chunk

       md13 : active raid1 sdb4[25]
                        458880 blocks super 1.0 [24/1]
[_U_____________________]
             bitmap: 1/1 pages [4KB], 65536KB chunk

       md2 : active raid1 sdb3[0]
             3897063616 blocks super 1.0 [1/1] [U]

       md256 : active raid1 sdb2[1]
             530112 blocks super 1.0 [2/1] [_U]
             bitmap: 0/1 pages [0KB], 65536KB chunk

       md9 : active raid1 sdb1[25]
                        530048 blocks super 1.0 [24/1]
[_U_____________________]
             bitmap: 1/1 pages [4KB], 65536KB chunk
```

As you can see from the above output, there is a disk with a 3rd partition that is most likely an MD LVM volume.
I'd say there is a 90% chance that this is the drive and partition we're interested in.

Take note of the device information, in this case **/dev/sdb3**

Invoke Testdisk Partiton Scan



So, again, we've detetermined the data to be on device **/dev/sda3**

The next step is to run testdisk against this device:

testdisk /dev/sdb3

In the ensuing dialog, choose the following order of actions:

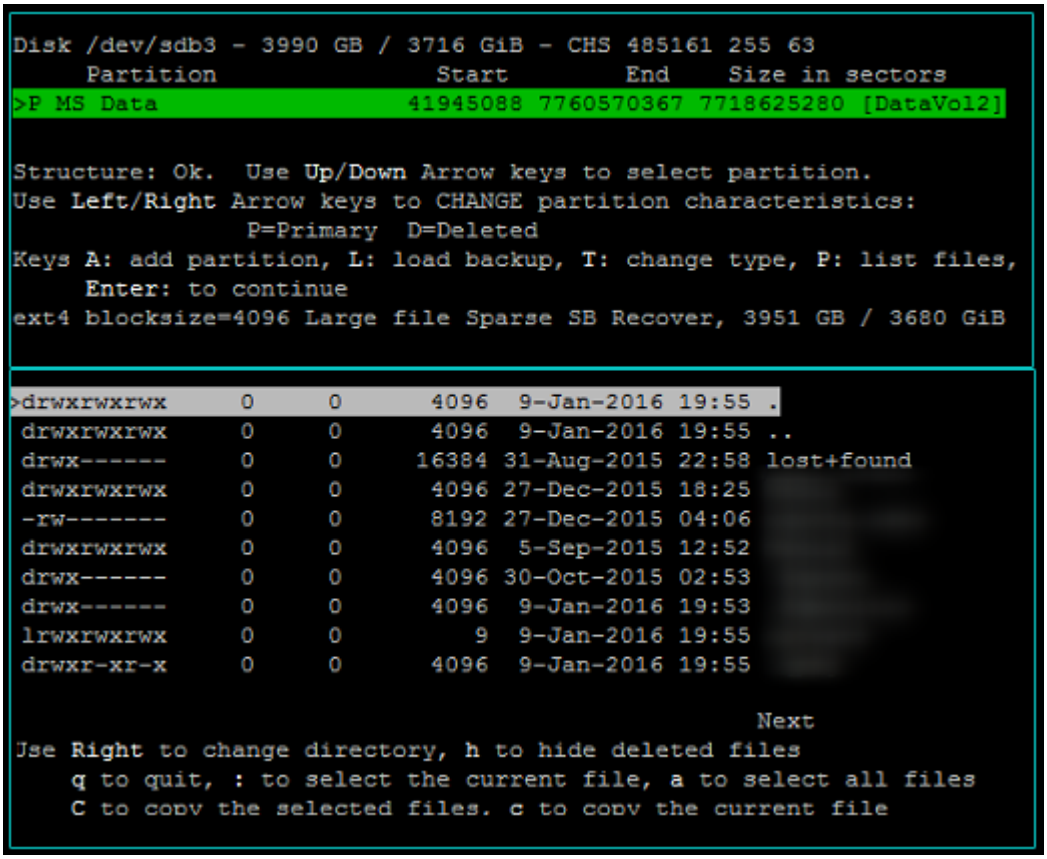
Select a media ...: **(choose the device, in this case /dev/sdb3)**

Proceed
Please select a partition table type ...: **(choose EFI GPT)**
Analyze
Quick Search

At this point, the drive scan will commence.

Once it completes, you'll be presented with a partition table as detected by testdisk.

List Files for Recovery & Copy



In the resulting partition table option, select the partition you think contains the data
Press **shift + P**
This will print the files on the partition
Read the instructions at the bottom of the file listing …

q to quit
: to select the current file
a to select all files

shift + C to copy the selected files
c to copy the current file

Once you invoke the copy action, you will be prompted to navigate to the destination path.

Hopefully you've already completed that in steps '**Prepare the Destination Drive**' and '**Mount the Destination Drive**'

Once the copy process is started, you'll be presented with a progress indication.

Sit tight. The wait is worth it.

Sources

[SMB] HOW-TO RECOVER data from LVM volume on a PC (UX-500P)
http://forum.qnap.com/viewtopic.php?t=93862

LAMP Stack with VirtualHosts On Centos 6.x

This article illustrates how to install the Apache Mysql PHP Stack on Centos 6.x.

Additionally, with this configuration, you can serve Multiple Domains using the Virtual Hosts Apache directive.

Install Apache

Invoke yum for installation of Apache

```
yum install -y httpd mod_ssl httpd-devel
```

@!:{httpd-devel libraries were included in order to have module compile capabilities, as well as being able to install modules from source
Enable autostart of the Apache service

```
chkconfig httpd on
```

Start the Apache service

```
service service httpd resart
```

Install PHP

Install PHP, et al

```
yum install -y php php-mysql php-common php-mbstring php-mcrypt php-devel php-xml php-pecl-memcache php-pspell php-snmp php-xmlrpc php-gd
```

Restart the Apache service

```
service httpd restart
```

Check DNS

Ensure there exists a DNS entry for the domain you want to use.

If this is a lab setup, or completely local, you can simply create a hosts entry for the domain, e.g.

```
vi /etc/hosts
```

[divider]

Virtual Hosts

The **NameVirtualHost** directive allows us to host multiple

websites on a single web server.

Example:

You want to host **mydomain1.com** on your web server
You also want to host **mydomain2.com** on your web server

In order to accomplish this, you'll need to:
– enable the NameVirtualHost directive
– create appropriate configuration files for the domains in question, e.g.:

```
/etc/httpd/conf.d/mydomain1.com.conf
/etc/httpd/conf.d/mydomain2.com.conf
```

For now, let's configure just one domain, *mydomain1.com*:

[divider]

Create Vhosts Config Directories

```
Create a vhost config folder

mkdir -p /etc/httpd/vhost.d
```

Configure NameVirtualHost Directive

Add an include directive to the apache config file:

```
vim /etc/httpd/conf/httpd.conf
    Include vhost.d/*.conf
```

@!:{The above makes it so that any files ending in .conf under the folder vhost.d are included as part of the httpd.conf configuration
Notice that ***vhost.d*** is a relative path. The full path would be evaluated as ServerRoot/vhost.d, where ServerRoot is

/etc/httpd (see the httpd.conf file for more information)

Comment out any Listen directives and add an include directive to a separate ports settings config file:

```
#Listen 12.34.56.78:80
#Listen 80
Include ports.conf
```

@!:{The above makes it so that the ports.conf file is included as part of the httpd.conf configuration
What this accomplishes is a separation of port specification from the main config file

Create a ports config file

```
vi /etc/httpd/ports.conf
```

With contents:

```
Listen $Port
NameVirtualHost $IPPUBLIC:$Port
NameVirtualHost $IPPRIVATE:$Port
NameVirtualHost *:$Port
```

Where **$Port** is the numeric value of the port number through which you want Apache to listen for traffic

```
#e.g.
NameVirtualHost 192.168.250.188:80
NameVirtualHost 127.0.0.1:80
NameVirtualHost *:80
```

Restart Apache

```
service httpd restart
```

Create The Config File for the Virtual Host/Domain

Create a config file for your domain

```
vim /etc/httpd/vhost.d/mydomain1.conf
```

```
<VirtualHost *:80>

ServerName mydomain1.com
ServerAlias www.mydomain1.com
DocumentRoot /var/www/vhosts/mydomain1.com
<Directory /var/www/vhosts/mydomain1.com>
Options Indexes FollowSymLinks MultiViews
AllowOverride All
</Directory>

CustomLog /var/log/httpd/mydomain1.com-access.log combined
ErrorLog /var/log/httpd/mydomain1.com-error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

</VirtualHost>
```

Make sure your document root exists!

```
mkdir /var/www/vhosts/mydomain1.com
#–OR Try this One-liner–#
ls /var/www/vhosts/mydomain1.com 2> /dev/null || echo does not exist;echo creating folder;mkdir -p /var/www/vhosts/mydomain1.com && echo created folder!
```

[divider]

Modify Firewall

You'll need to poke a hole in the firewall to allow communication to the Apache listening port (by default port 80):
Edit iptables config

```
vi /etc/sysconfig/iptables
A INPUT -m state –state NEW -m tcp -p tcp –dport 80 -j ACCEPT
```

Restart iptables

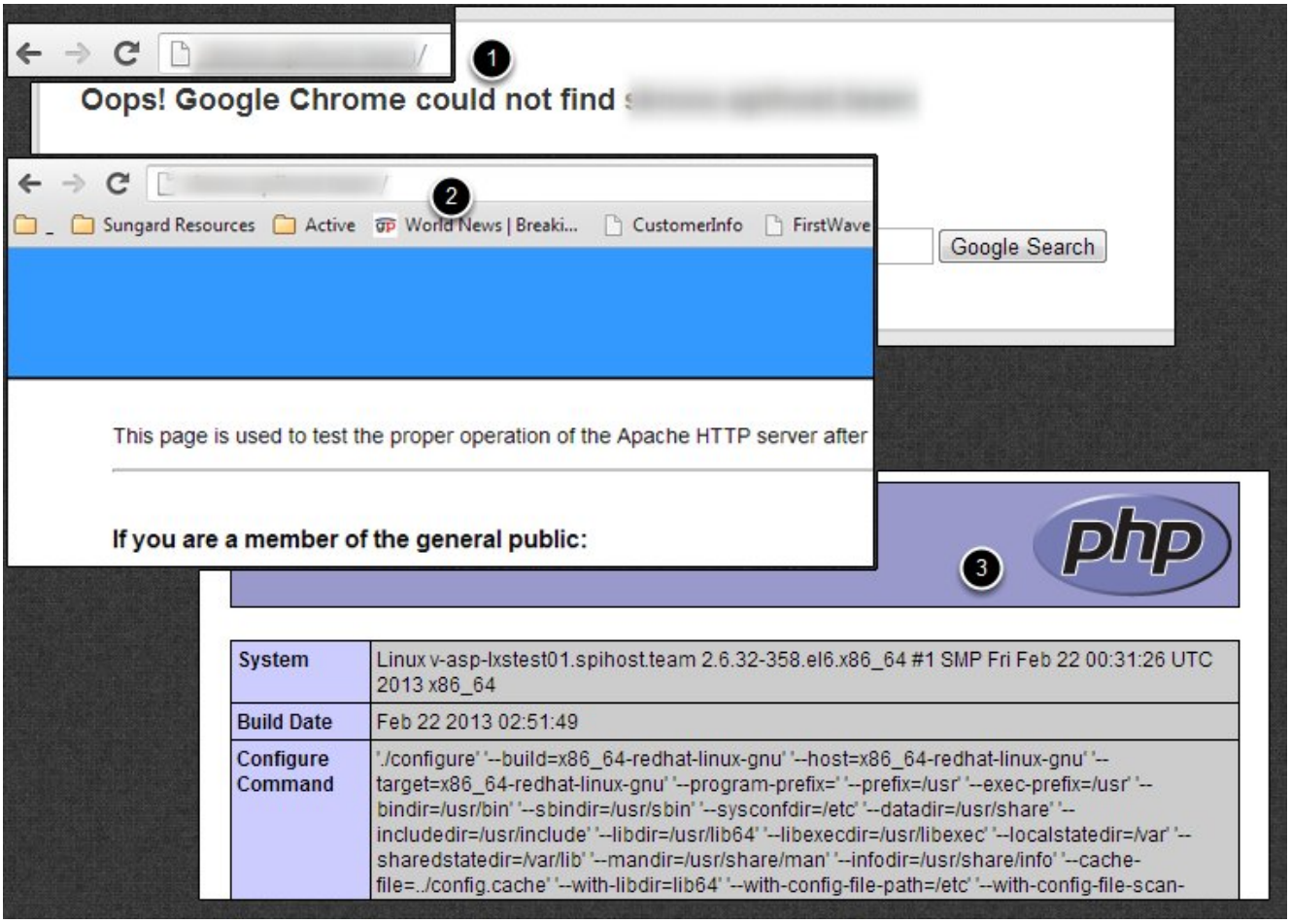
```
service iptables restart
```

[divider]

Troubleshooting

[divider]

Error – Could not find …



1. Problem: When navigating to your domain via web browser, you receive an error similar to ‘could not find’

Q:{Is DNS setup correctly?
Check:

```
nslookup mydomain1.com
```

if error then ensure DNS record exists on your DNS server

if Windows, try the *ipconfig /flushdns* command

Q:{Is Firewall to blame?
Check:

```
telnet $yourdomain $port
```

e.g.

```
telnet mydomain1.com 80
```

if error then ensure Firewall port is open:

```
vi /etc/sysconfig/iptables
e.g. -A INPUT -m state –state NEW -m tcp -p tcp –dport 80 -j ACCEPT
```

Restart firewall:

```
service iptables restart
```

2. Test website access again
Hopefully Success!

3. Test PHP functionality:

```
vi /var/www/vhosts/domain.com/index.php
```

```
<?php
phpinfo();
?>
:wq
```

Test website access again

```
http://mydomain1.com/index.php
```

If you've made numerous changes, try restarting the Apache service again

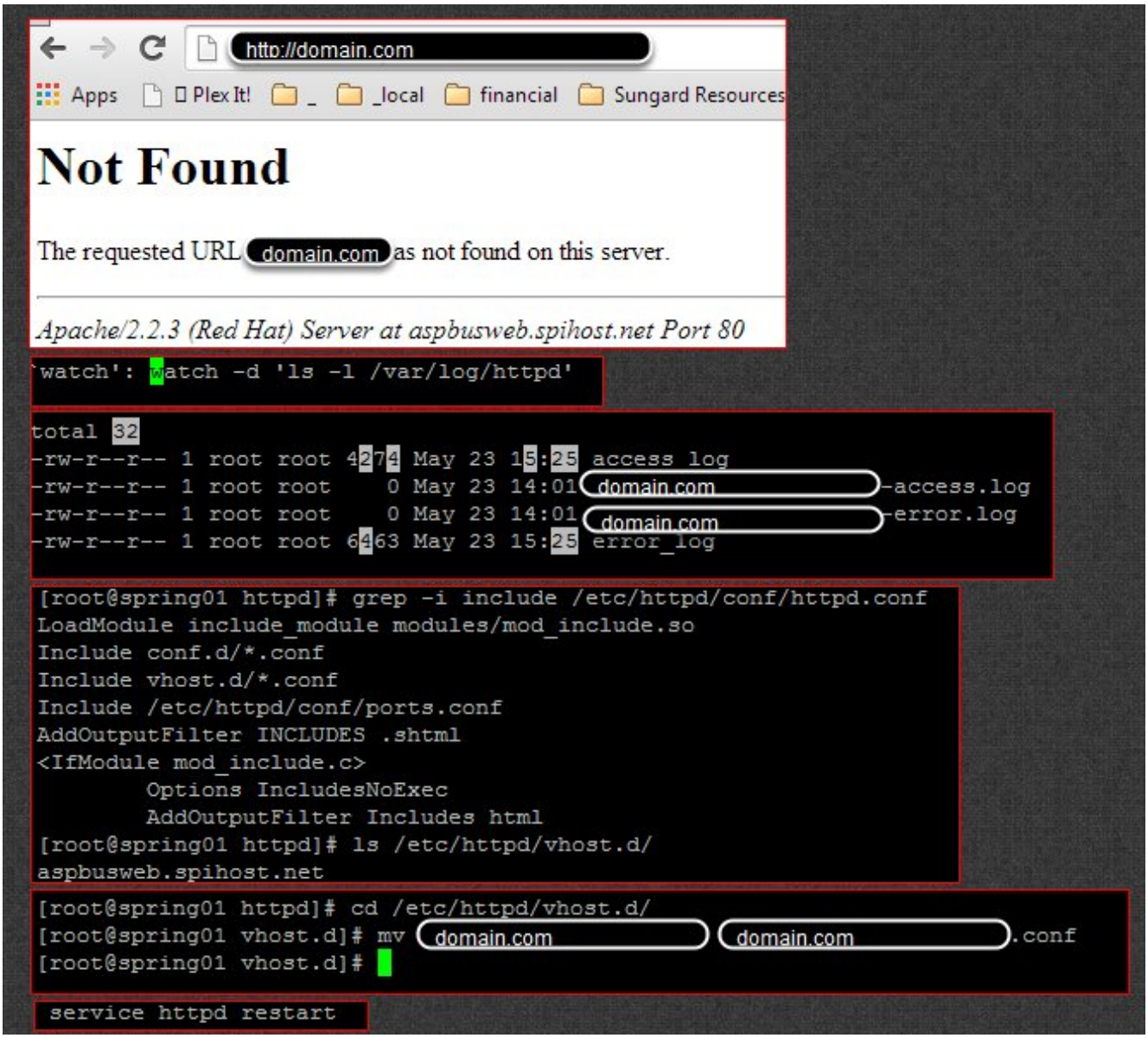
```
service httpd restart
```

If all else fails, and if you have the option to do so, reboot the server

```
reboot
```

Error – requested URl was not found on

this server



In this case, I created the config file for the domain under vhosts.d, but had forgotten to give it a .conf file extension. doh!

Note how I used the watch command to ‘watch’ for changes to log files under /var/log/httpd.

This functions much like inotifywait for troubleshooting using log files.

Adding a Network Card to CentOS Linux

Detect & Configure The New Network Adapter

1. Determine existing network interfaces

ifconfig -a
2. Change directory to the network scripts folder
cd /etc/sysconfig/network-scripts
3. Clone the existing eth0 device network script
cp ifcfg-eth0 ifcfg-eth1 # this assumes the old card was eth0 and the new one is eth1
4. Get the Hardware Address for the eth1 network card, again this assumes the new card is eth1
grep eth1 /etc/udev/rules.d/70-persistent-net.rules
#you can get fancy and use awk and cut to isolate the string containing the Hardware Address
grep eth1 /etc/udev/rules.d/70-persistent-net.rules | awk -F"," '{print $4}' | cut -d= -f3
5. Replace all occurences of eth0 with eth1 in the new network configuration script
sed -i 's/eth0/eth1/g' ifcfg-eth1 # or edit it by hand and change eth0 to eth1 where it appears
6. Edit the eth1 network configuration script and replace the Hardware Address with the one in the 70-persistent-net-rules file
vi ifcfg-eth1
7. Bring the eth1 interface up
ifup eth1