Sentinel Data Security

DISK CLONING

Mario Horvat

Sentinel Data Security

<marioh_no_spam@sentinelsecurity.net>
Key fingerprint = 5B0C 0342 0370 7A81 8876 F33D 3B32 AB46 06F5 4041

Revision history

Revision v1.8	15 January 2005
Revision v1.7	27 February 2004
Revision v1.6	28 November 2003
Revision v1.5	18 November 2003
Revision v1.4	25 September 2003
Revision v1.3	12 August 2003
Revision v1.2	19 November 2002
Revision v1.0	21 July 2002
Revision v0.1	3 May 2002

Table of Contents

1. About this document

When partitioning a hard drive we are faced with a choice of: 4 primary partitions only or 3 primary + extended (extended will consists of logical drives). If we were to create 4 primary partitions, all the partition information will be stored in the first 512 bytes of the disk. However, in most cases having only 4 partitions poses a limitation on what we can do with the system. In this case we need to create an extended partition.

An extended partition is a primary partition with a twist; in a sense that all the logical drivers are stored in the form of a linked list. When creating a logical partition we have the first 3 partitions being the primary partitions and 1 being an extended partition.

This document describes how to successfully clone a disk whether it contains all primary partitions or a mix of both primary and extended partitions.

2. Copyleft and License

3. Partition table history

Partition tables are stored in partition boot sectors. Usually the partition boot sector of the disk is used as the boot sector, this is also known as the master boot record. Its structure, borrowed from (www.mobiledyne.com/pub/mbrh.html) is as follows:

```
struct master_boot_record {
  char    bootinst[446];
  char    parts[4 * sizeof (struct fdisk_partition_table)];
  ushort  signature; /* 0xAA55 */
};
```

```
          MBR
          +--------------------------------+
0x000     |
          |
          |
          |      Program code
          |
          |
          |
          |--------------------------------|
0x1BE     |
          |      Partition table
          |
          |
          |
          |--------------------------------|
0x1FE     | Magic number (0xAA55)
          +------------------------------+
```

4. LFS Debate

I have received quite a few emails about LFS support and why people can’t create files >2Gb when creating partition images. First of all, the problem is not just in the kernel support. Ever since kernel 2.4.0-test7 the interfaces for LFS are built into the kernel. The problem also lies in the binaries which are compiled with or without the LFS APIs. Most recent distributions do not have this problem and I suggest that you look at the distribution you are using. More information about this can be found here: http://www.suse.de/~aj/linux_lfs.html

5. Tools

Things we need:

- dd
- sfdisk
- netcat/cryptcat/scp (for remote cloning)
- larger or identical disk on destination host (for local cloning)
- boot disk/cdrom such as fire or knoppix

6. Useful links

Fire	http://fire.dmzs.com/
Knoppix	http://www.knoppix.net/
Tom’s root/boot disk	http://www.toms.net/rb/
MBR info	http://www.mobiledyne.com/pub/mbrh.html
LFS information	http://www.suse.de/~aj/linux_lfs.html

7. Clone local

NOTE: Make sure you have an understanding of how dd works before you embark on this adventure.

For example, this procedure is designed to clone a running system. So we are able to clone a disk on a running mission critical server (with a few files having a small difference eg. syslog), without having to down the system.

This means that we will have to be careful where we are putting the images. Ie. In case of a network restore you can't backup everything on one partition and then dd that partition to itself. Best way (we found) to do this is probably by **choose**(ing) the **largest partition on** the **system**, **backup all other** partitions to it, **then after transferring** the **images**, **remove them** from the local disk and **backup** the **last partition to** one of the **other** partitions and **then transfer it** across. Even better, use something like cryptcat or netcat via ipsec and do it on the fly.

The disks used here are:

- Maxtor D740X-6L 20Gb
- Seagate Barracuda ST340016A 40Gb

Using disks hda (20Gb primary master) and hdc (40Gb secondary master) for example cloning from hda to hdc. Note that this method is not restricted in any way and may have a lot of variations. For example backing up locally, burning to DVD then restoring remote in which case you might restore to hda again. Cloning a drive to another can simply be done by the use of a single *nix command and does not require any more steps than that. This document however, covers the procedure for cloning a disk on partition by partition basis. This may or may not be useful for some.

Let us first examine the disk which we will use to clone from:

Using fdisk:

```
root@from# fdisk –l /dev/hda
```

```
Disk /dev/hda: 255 heads, 63 sectors, 4865 cylinders
Units = cylinders of 16065 * 512 bytes

  Device Boot   Start    End    Blocks   Id  System
/dev/hda1   *       1      4    32098+  83  Linux
/dev/hda2           5     40   289170   83  Linux
/dev/hda3          41     64   192780   83  Linux
/dev/hda4          65    143   634567+   5  Extended
/dev/hda5          65     88   192748+  83  Linux
/dev/hda6          89    112   192748+  83  Linux
/dev/hda7         113    143   248976   82  Linux swap
```

Using sfdisk:

```
root@from# sfdisk –d /dev/hda
```

```
# partition table of /dev/hda
unit: sectors

/dev/hda1 : start=       63, size=   64197, Id=83, bootable
/dev/hda2 : start=    64260, size=  578340, Id=83
/dev/hda3 : start=   642600, size=  385560, Id=83
/dev/hda4 : start=  1028160, size= 1269135, Id= 5 (extended partition)
/dev/hda5 : start=  1028223, size=  385497, Id=83
/dev/hda6 : start=  1413783, size=  385497, Id=83
/dev/hda7 : start=  1799343, size=  497952, Id=82
```

Just so you are familiar with what kind of partitions we will be dealing with. These will of course be different on your system, unless by some pure coincidence you chose the same partition layout and sizes. Not likely.

Step 1 (backing up the partition information):

This section deals with cloning a disk to another disk on the same system. This will mean that the disk will have to be physically plugged into the system we wish to clone. This can usually be done with a straight dd mirror.

NOTE: there is always an option of doing a compressed backup. This of course if we are doing the restore off the network is going to be very useful in both time and cost savings. In this case we can pipe to gzip to produce compressed images. You can use some other compression if you like. I used gzip.

```
root@from# dd if=/dev/hda of=/safe/place/hda.mbr.dd bs=512 count=1
root@from# sfdisk –d /dev/hda > /safe/place/hda.pt.sfdisk
```

Step 2 (backing up the partitions with dd):

```
root@from# dd if=/dev/hda1 bs=512 | gzip –9 > /safe/place/hda1.dd.gz
root@from# dd if=/dev/hda2 bs=512 | gzip –9 > /safe/place/hda2.dd.gz
root@from# dd if=/dev/hda3 bs=512 | gzip –9 > /safe/place/hda3.dd.gz
root@from# dd if=/dev/hda5 bs=512 | gzip –9 > /safe/place/hda5.dd.gz
root@from# dd if=/dev/hda6 bs=512 | gzip –9 > /safe/place/hda6.dd.gz
```

NOTE: bs=512 is the default. If the sector size of the disk is not 512, bs should be set to a multiple of that sector size. We do not need to back up **hda4**, that is the extended partition and sfdisk will take care of that.

Now that we backed up our partitions we need to restore it to the clone disk.

Step 1 (restore the partition information):

```
root@to# dd if=/safe/place/hda.mbr.dd of=/dev/hdc bs=1
```

NOTE: we are restoring the information from **hda** to our clone disk **hdc**.

First we test the extended partition information which we will be writing to the new disk with sfdisk. The –n switch does not allow writing to the disk. Instead it only tests the configuration.

```
root@to# sfdisk –n /dev/hdc < /safe/place/hda.pt.sfdisk
```

If everything goes well we can proceed to actually writing to the disk.

```
root@to# sfdisk /dev/hdc < /safe/place/hda.pt.sfdisk
```

NOTE: We need to make sure the kernel knows about the new disk configuration. To make sure that the kernel has updated this information:

```
root@to# sfdisk –R /dev/hdc
```

Step 2 (restore the partitions):

```
root@to# gzip –c –d /safe/place/hda1.dd.gz | dd of=/dev/hdc1 bs=512
root@to# gzip –c –d /safe/place/hda2.dd.gz | dd of=/dev/hdc2 bs=512
root@to# gzip –c –d /safe/place/hda3.dd.gz | dd of=/dev/hdc3 bs=512
root@to# gzip –c –d /safe/place/hda5.dd.gz | dd of=/dev/hdc5 bs=512
root@to# gzip –c –d /safe/place/hda6.dd.gz | dd of=/dev/hdc6 bs=512
```

8. Clone remote

This section deals with cloning to a remote system either over the LAN or the Internet/WAN assuming there is a direct link between the two machines and that at least one port (port 2000 in this example) is available through the firewalls, hence direct connection.

This part involves sending and receiving the data over a network, how its done is up to you, we will only cover the procedure using netcat. You can use ssh or whatever other means you wish. I will use from/to terminology to distinguish between the backup and restore machines. “**from**” is obviously the machine which has data which we wish to clone, and “**to**” is obviously a machine which is going to be used to store the data.

Step 1 (restore the partition information):

Using netcat to listen on port 2000 and dump straight to the restore disk.

```
root@to# nc –v –l –p 2000 | gzip –cd > /dev/hda
```

Set up a listener on the destination box port 2000 to dump straight to the disk.

```
root@from# dd if=/dev/hda bs=512 count=1 | gzip –c9 | nc –q 2 <IP> 2000
```

We dd the mbr (first 512 bytes) gzip it on the fly and pipe it to the remote machine.

NOTE: we are restoring the **hda** information to another **hda** on a remote box who’s IP is <IP>.

Because I didn’t have any luck with piping straight to sfdisk (let me know if you did) I dumped the information into a file first and then tested the config with it.

So we set up another listener, this time for the sfdisk info.

```
root@to# nc –v –l –p 2000 | gzip –cd > hda.sfdisk
```

This sets up a listener and will pipe the info to a file called hda.sfdisk

```
root@from# sfdisk –d /dev/hda | gzip –c9 | nc –q 2 <IP> 2000
```

This dumps the extended partition information to gzip which compresses it on the fly and finally connects to the remote IP and dumps it.

Now that we have the extended partition information in the file hda.sfdisk on the remote machine we can proceed.

```
root@to# sfdisk –n /dev/hda < hda.sfdisk
```

This will attempt to test whether the extended partition info fits in with the mbr and the size of the disk.

The –n switch does not allow writing to the disk. Instead it only tests the configuration. If everything goes well and we get no errors we can proceed to actually writing to the disk.

```
#sfdisk /dev/hda < hda.sfdisk
```

Step 2 (restore the partitions):

Now that we are done with all this mbr reconstruction we can proceed to dump the partitions in their correct place.

```
root@to# nc –v –l –n 2000 | gzip –cd | dd of=/dev/hdaX
```

Set up the listener

```
root@from# dd if=/dev/hdaX | gzip –c9 | nc –q 2 <IP> 2000
```

Dump the partition.
Repeat this step for the number of partitions that you have. All that will vary is the partition **X** eg. hda2, hda3, hda4 etc.

NOTE: we will have to do the partition dumping one at a time, depending on the size of your partitions this process could take a while. What I have noticed with this on the fly compression is that it is not as good as using gzip or bzip2 to compress the whole images. If traffic is an issue for you, ie you are paying for bandwidth; you might wish to consider backing up, compressing and then sending the images.

9. Quick reference guide

This is a step by step solution for the people who want to get straight to the point. You can print only this page and use it as a quick reference.

Local Step 1:

```
root@from# dd if=/dev/hda of=/safe/place/hda.mbr.dd bs=1 count=512
root@from# sfdisk –d /dev/hda > /safe/place/hda.pt.sfdisk
root@from# dd if=/dev/hda1 of=/safe/place/hda1.dd bs=512
root@from# dd if=/dev/hda2 of=/safe/place/hda2.dd bs=512
root@from# dd if=/dev/hda3 of=/safe/place/hda3.dd bs=512
root@from# dd if=/dev/hda5 of=/safe/place/hda5.dd bs=512
root@from# dd if=/dev/hda6 of=/safe/place/hda6.dd bs=512
```

Local Step 2:

```
root@to# dd if=/safe/place/hda.mbr.dd of=/dev/hdc bs=1
root@to# sfdisk –R /dev/hdc
root@to# sfdisk –n /dev/hdc < /safe/place/hda.pt.sfdisk
root@to# sfdisk /dev/hdc < /safe/place/hda.pt.sfdisk
root@to# dd if=/safe/place/hda1 of=/dev/hdc1 bs=512
root@to# dd if=/safe/place/hda2 of=/dev/hdc2 bs=512
root@to# dd if=/safe/place/hda3 of=/dev/hdc3 bs=512
root@to# dd if=/safe/place/hda5 of=/dev/hdc5 bs=512
root@to# dd if=/safe/place/hda6 of=/dev/hdc6 bs=512
```

Remote Step 1 (clone the mbr)

```
root@to# nc –v –l –p 2000 | gzip –cd > /dev/hda
root@from# dd if=/dev/hda bs=512 count=1 | gzip –c9 | nc –q 2 <IP> 2000

root@to# nc –v –l –p 2000 | gzip –cd > hda.sfdisk
root@from# sfdisk –d /dev/hda | gzip –c9 | nc –q 2 <IP> 2000

root@to# sfdisk –n /dev/hda < hda.sfdisk
Test OK…
root@to# sfdisk /dev/hda < hda.sfdisk
root@to# sfdisk –R /dev/hda
```

5.4 Remote Step 2 (clone the partitions):

```
root@to# nc –v –l –p 2000 | gzip –cd | dd of=/dev/hdaX
root@from# dd if=/dev/hdaX | gzip –c9 | nc –q 2 <IP> 2000
```

Repeat this step for the number of partitions that you have. All that will vary is the partition **X** eg. hda2, hda3, hda4 etc.

Appendix A: about Sentinel Data Security

Established in 1998, Sentinel Data Security is an information security consultancy. It is a leading supplier of value for money threat management services, helping organisations develop, implement, and maintain best practice business security strategies for their information assets.
Sentinel Data Security take a complete approach to data security, offering a variety of consulting and solution services, with defined deliverables to suit your organisations needs.
Its people consult to a diverse range of entities; from small businesses to blue chips, and government departments & agencies. These engineers and consultants have broad experience gained over years in the security arena, and backed up by official certification from security industry bodies. It is this experience that makes these consulting services relevant, and therefore invaluable to Sentinel clients. Quality management and standards are implemented through Sentinel Data Security's 'Security Management Framework'. This policy and procedures document has been developed specifically for Sentinel Data Security and has been closely aligned to AS/NZS 7799.2:2000 standards.
Sentinel's dedicated and talented experts, quality infrastructure and management services ensure your security needs will always be in good hands. Sentinel can help your organisation in the following areas:

consulting services

- **information security governance**
- **strategy development**
- **roles, responsibilities and reports identification**
- **policies, procedures and guidelines development**
- **national privacy policy due diligence auditing**

- **risk management**
- **process development**
- **risk identification and analysis**
- **risk mitigation strategy development**

- **information security programme management**
- **security governance framework implementation**

- **information security management**
- **compliance assessment**
- **metrics development**
- **balanced scorecard development**
- **framework development**
- **wireless system policy, auditing, planning and design**

- **response management**
- **root cause analysis**
- **mitigation strategy development**
- **backup strategy development**
- **business continuity planning**
- **disaster recovery planning**
- **forensics**
  - anton pillar search orders
  - network / system forensics
- **vulnerability assessment**
  - penetration testing
  - social engineering
  - application source code audits
  - network / application design audits

solutions

- **design & implementation of ids, firewalls, vpn's**
- **perimeter security**
  - firewalls
  - VPN's
  - load balancing
  - traffic shaping
  - honey nets
- **intrusion detection**
  - network intrusion detection systems
  - host intrusion detection systems

- **design & implementation of authentication & encryption**
- **authentication solutions**
- **encryption solutions**
  - PGP
  - PKI

- **design & implementation of content security**
- **content access control**
- **post delivery policy control for email & documents**
- **content filtering & logging**
- **virus control**

- **tools**
- **self auditing software**

24 x 7 Managed Services

- **intrusion detection system monitoring**
- **intrusion detection system management**
- **firewall monitoring**
- **firewall management**
- **sms & paging alerts**
- **authentication token management**
- **document and email security management**