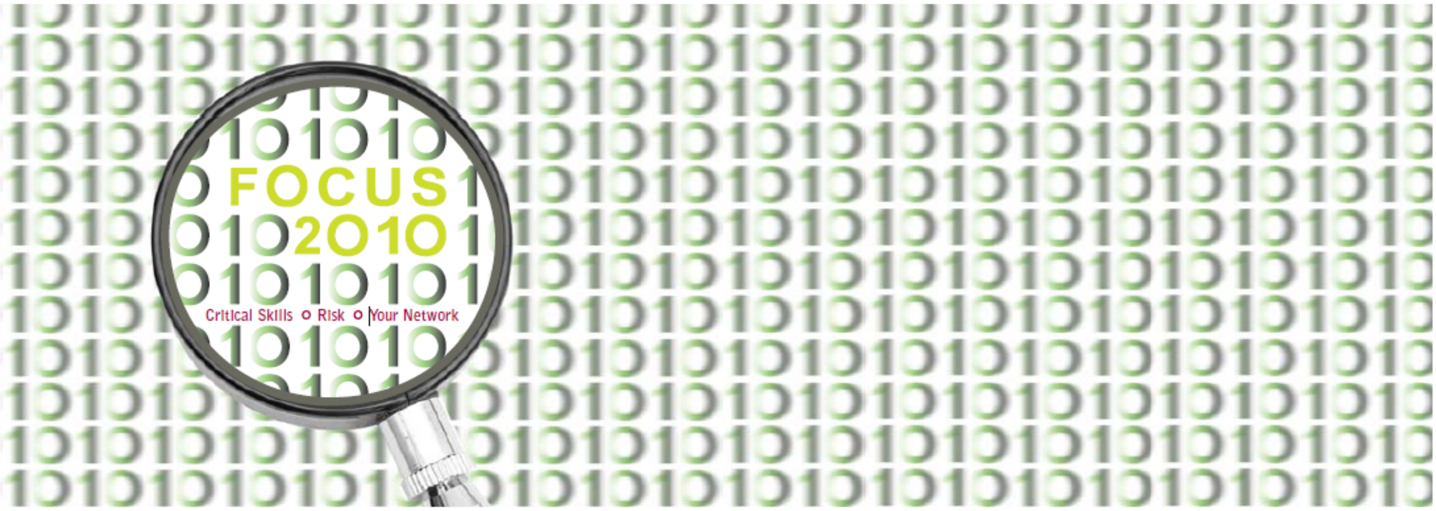
10th Annual SF ISACA Fall Conference

October 4 – 6, 2010



S23: You Have Been Hacked, But Where's the Evidence? A Quick Intro to Digital Forensics

Bill Pankey, Tunitas Group

ISACA®
Trust in, and value from, information systems
San Francisco Chapter

You Have Been Hacked, But Where's the Evidence? A Quick Intro to Digital Forensics

Bill Pankey
CGEIT, CISA, CISSP, GCFA, GCIH, GCUX, MCSE, QSA
Tunitas Group

Goals

Provide a first introduction to digital forensics

- Problem addressed
- Issues encountered
- Tools Used
- Acceptance of results

[*very*] Lightly scratch surface to demonstrate tools in the investigation of an incident

Biased toward the use GNU (free) tools

- No comments on proprietary products

2 Views of Computer Forensics

1. the use of science and technology to investigate and establish facts in criminal or civil courts of law.
   - *litigation & criminal prosecution*
2. the study of evidence from attacks on computer systems in order to learn what has occurred, how to prevent it from recurring, and the extent of the damage. (NIST SP800-86)
   - *problem management*

Differing Focus

collection of evidence related to:

1. a *crime or claim* where the electronic aspect is only circumstantial
   - seek an *engineering* solution for the legal problems of the reliability | relevance of evidence
2. the *unexpected behavior of the system,* i.e. electronic aspect is central
   - make conclusions on the basis of the state of the [un-trusted] targeted system | networks

In every case …

1. Gathering of data / evidence
   - ‘Opportunity costs’ if data not collected as early as possible.
   - Time is of essence, but need for speed can be overstated
2. Investigation and analysis
   - Data recovery, timeline and event reconstruction
3. Reporting
   - Description of what occurred in a way that compels some action

Evidence Gathering | Acquisition

Principles

- Minimize loss of data
- Preserve integrity of data
- Document everything
- Have a plan
  - *policy guidance ?*

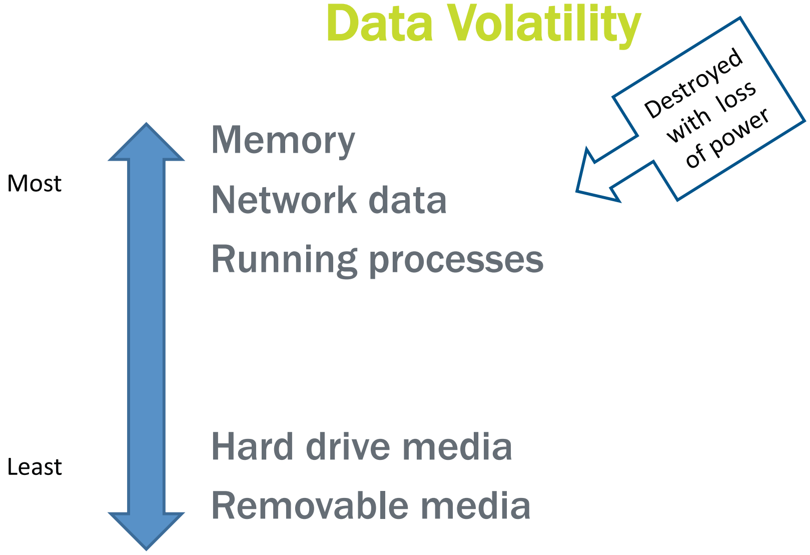
Planning Considerations

- Volatility of data
- Likely value of data
- Level of effort to recover

Challenges

1. Rapid [deliberate] action
   – volatility of system information and data
2. Collecting system data w/o altering the system
   – Dependent on the execution of software
3. Trusting the software tools
   – what is to be trusted in a compromised system
4. Preserving and demonstrating evidence integrity
   – may need to argue the negative

Data Volatility



Memory

Network data

Running processes

Hard drive media

Removable media

Volatile data is lost as a result of actions performed on the system <= time

DEMO

An Incident reported by network monitoring tool

- ‘unexplained’ transfer of files

Involves a Hacked [TBD] Windows 2003 server

- ‘Hacker Defender’ Rootkit
- hidden Netcat

What is learned?

- Running system utilities from a command line?
- Same tools run from a CD?

Contamination Issue

Possible to acquire much of desired data by running tools from ‘trusted’ sources, *but*

- use of tools runs in process and affects memory
- does not preserve the evidence

Better ….

1. *Capture memory to an image file using tool*
2. *Protect image file*
3. *Extract desired data from memory image*

Conventional forensic methodology used regardless of target

Demo: Memory Capture

Win32dd.exe usageSwitches:

/f <file> destination file

/

Capturing Memory (Tools)

Windows

Live system

- win32dd.exe
  - http://win32dd.msuiche.net/
  - Computes MD5 hash of memory image file
- mdd.exe
  - http://www.mantech.com/msma/mdd.asp
- Memoryze
  - http://www.mandiant.com/software/memoryze.htm

Dead system

- Compressed RAM in Hibernation File (`hiberfil.sys`)
- Windows Memory Toolkit
  - http://www.moonsols.com/component/jdownloads/view.download/3/2

What is in Memory?

Memory includes

- running Processes
- open connections; listening ports
- open files
- configuration parameters
- encryption keys
- *interesting* text data

But , what information do we want from memory?

- incident verification / identification?
  - Backdoors, hidden files, unusual processes, …
- Queries that rise during the investigation?

strings

Utility that outputs strings of printable characters in binary files

- Linux / unit command
- Windows executable from sysinternals

Pipe *strings* output to *grep* / *wingrep* searching for ‘dirty word’ occurrences

Usefulness of such search depend upon type and specifics of the incident

- Iterate during analysis phase once the image is obtained

volatility

Perl script for analysis of windows memory image files

- list connections
- list running processes
- list open files by process
- list open registry keys in each process
- list open sockets
- dump process to an executable

DEMO

#python volatility <command> -f <path to image>

- *pslist* command: list running processes
- *connscan* command: connection objects

Does volatility output confirm incident?

- likely rootkit
- hacker’s access code

What next?

- isolate system? disconnect from network?
- ‘pull the plug’ ?

Volatile Data: Best Practice

- ☼ Obtain memory image
  1. Minimize any other action on suspect machine
  2. Run acquisition tool from external media or network source
  3. Pipe output to external media
  4. Protect image file

Confirm incident

Take appropriate incident response steps

Contain (Disconnect / UnPlug)

Eradicate ???

Note about Timing

Urgency is reduced once volatile data is collected:

- Memory
- Time line data

Further data acquisition and analysis can proceed at more relaxed pace

- Relatively little additional opportunity cost due to latencies in response

Truth be told, compromise to discovery typically measured in months .... from the 2009 Verizon Report

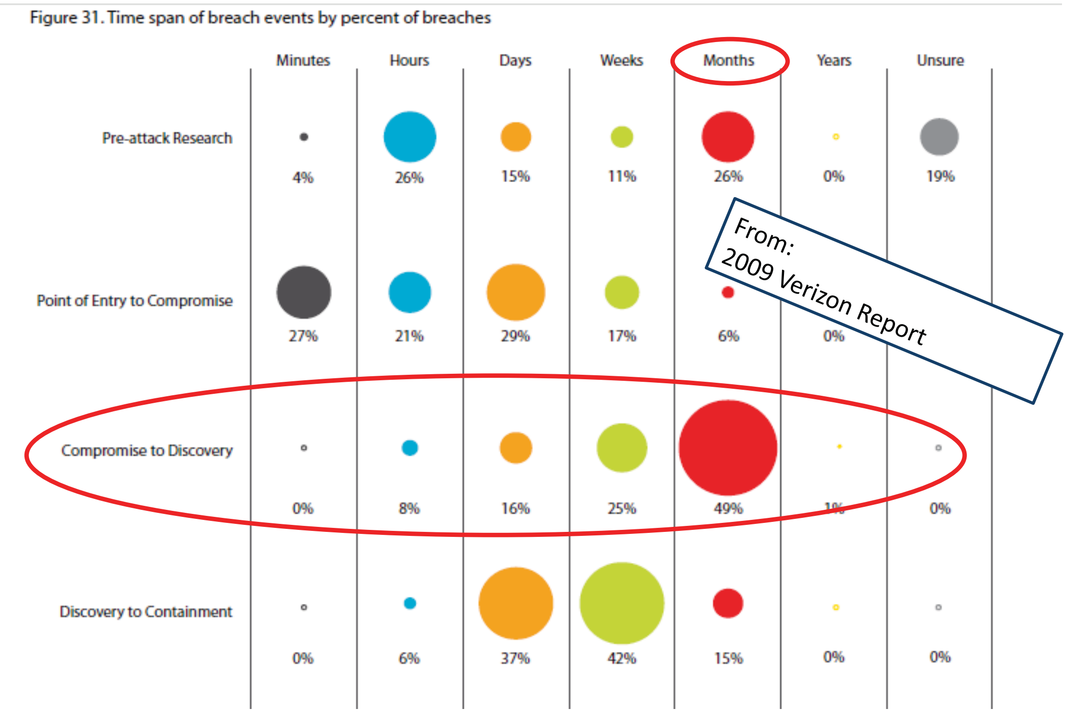


Figure 31. Time span of breach events by percent of breaches

From:
2009 Verizon Report

Media (Disk) Analysis

Persistent artifacts of the incident are on disk

- malware
- illicit or other unauthorized data
- incriminating evidence
- hidden logs and history files

Normally, the principal forensic focus

- generally the richest data source
- contains history of the incident
- contains the configuration data underlying the vulnerability that lead to the incident

Media Analysis

Similar Approach

- acquire and protect an accurate image file
- search the image for artifacts of the incident
  - ‘dirty words’
  - suspect files
  - hidden files
- recover data
- construct timelines
- correlate with other evidence

Caveats

Media analysis is time & resource consuming

- analysis should have defined objectives
  - where is the rootkit hidden?
  - what other files did the rootkit contaminate?
  - when was the system compromised?
  - what accounts were involved in the incident?
  - who was involved in the incident?

Media analysis will yield fragmentary evidence

- timeline data is overwritten with access to disk files
- slack space & deleted files are [eventually] overwritten

Requirement to correlate results with other evidence

Disk Imaging

Objective

- Create an 'authentic' copy of the disk as of a specific date / time
  - ○ Demonstrate its authenticity
  - ○ Ensure that analysis will be complete (deleted files, hidden data, slack space)

Best practice

- Prevent any further writes \ changes to disk
  - ○ ie, mount 'read-only'
- Obtain a bit-wise copy of the disk
- Copy to a standard format for use by analysis tools
  - ○ Raw (DD): original bit image; no metadata
  - ○ Advanced Forensic Format (AFM): DD + second file w/ metadata
  - ○ Expert Witness Format (EWF); Encase format, compression, metadata

Common Media Acquisition

1. Boot from CD
   - Helix or other linux
2. Mount target disk READ only
   - Evidence will remain in static state
3. Create bitwise copy
   - Image disk to a file on an attached device
     - ○ USB / Firewire connected large capacity drive, or
     - ○ Across network to analysis workstation
4. Mount image for analysis
5. Recover file system

dd

dd if=<*device name*> of=<*output file*> {options}

- Windows and Unix / Linux versions
- Computes md5 or image file and verifies against original

Alternative image collection

- Acquire image across network
  - Before powering off, mount the drive read_only as an external device on a connected machine and then copy
- Remove drive and connect to analyst workstation with a USB to IDE / SATA drive adapter

DEMO

Understanding the Image: File System Basics

Knowledge of file systems is required to:

- construct accurate, if [potentially] incomplete timeline of file related events
- recover information in files w/o having to rely upon OS mechanisms
  - Preserving the integrity of evidence
  - Avoid reliance upon potentially compromised system
- find information in deleted files
- find information in slack space (i.e., space unused by or otherwise invisible to OS)

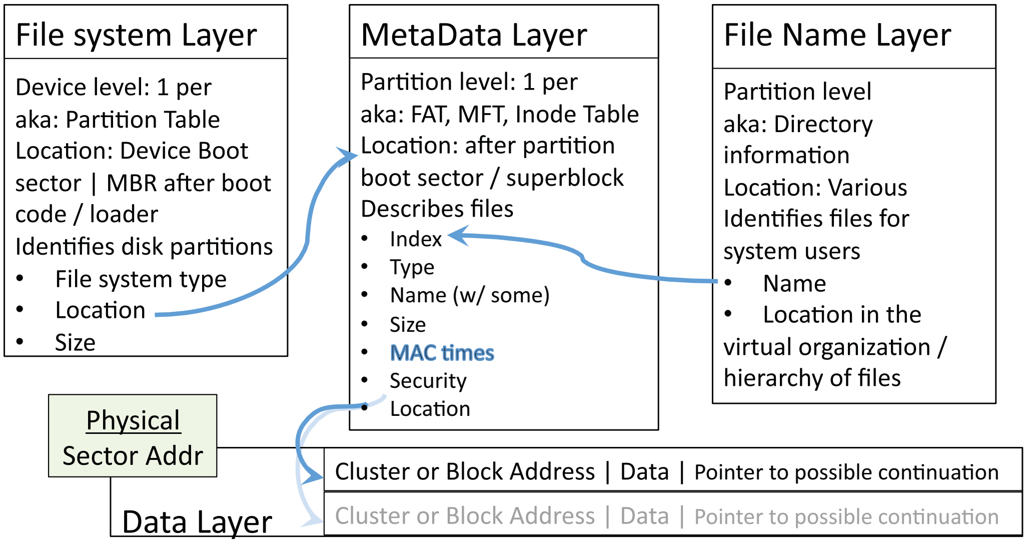
File System Types

File systems generally intended to support different OS / applications

Analysis tools must be able to recognize layouts

File System	Targets
FAT	MSFT
NTFS	MSFT
exFAT	USB flash, Windows CE
Ext2, ext3	Linux
ext4	Linux
ACFS	Oracle Enterprise Linux
UDF	ISO / ECMA
Resier	Linux
Google File System	Google
HFS	Mac
HFS	MVS / zOS
zFS	z/OS
VMFS	Vmware
ZFS	Solaris, BSD

Common Disk Arrangement (Logical)



File system Layer

Device level: 1 per
aka: Partition Table
Location: Device Boot sector | MBR after boot code / loader
Identifies disk partitions

- File system type
- Location
- Size

MetaData Layer

Partition level: 1 per
aka: FAT, MFT, Inode Table
Location: after partition boot sector / superblock
Describes files

- Index
- Type
- Name (w/ some)
- Size
- MAC times
- Security
- Location

File Name Layer

Partition level
aka: Directory information
Location: Various
Identifies files for system users

- Name
- Location in the virtual organization / hierarchy of files

Physical Sector Addr

Data Layer

Cluster or Block Address | Data | Pointer to possible continuation

Cluster or Block Address | Data | Pointer to possible continuation

MetaData Layer

Collection of data structures that describe the file, aka:

- Inode table (Unix / Linux)
- Master File table (NTFS)
- File Allocation Table (FAT)

1 data structure per file

- Pointers to file content (data)
- MAC Times
- Permissions

metaData: TimeStamps

“MAC” times are essential to investigation

- basis for event correlation

Meaning of entries changes with file system type

	M (data layer)	A (data layer)	C (meta data layer)	B (‘birth’)
EXt2/ Ext3	Modified	Accessed	Inode Change	
FAT	Modified	Access Date		File Creation
NTFS	Modified	Accessed	MFT Change	File Creation

- If file deleted, C= deletion time
- if, M < C => file has been copied

○ Only the last time is preserved

Timeline Analysis

Inspect system activity around time of incident

- Files that were accessed, deleted or changed
- Tool execution
- Patterns
- Recent modification in system / critical files

Timelines are some of the more reliable evidence:

File system maintains timeline, so difficult for a rootkit to manipulate; MAC data for files as well as commands that hide processes

but, very sensitive to changes in system so must be captured early

Search for Dirty Words

Presence of certain text may provide evidence

1. Use Strings to create file of all text stings in binary file
2. Use grep | wingrep | search strings or similar to find targeted terms

Meaningful ‘dirty word’ list is case specific, eg:

- Eg, regular expression for SSN or bankcard # when investigating workstation owner for theft and misuse of PII
- IP or URL
- file names

Check File Integrity

Mis-match of a file's hash and the "known_good' value may indicate tampering

Known_good

- Hash database of system files computed prior to incident
  - ○ MD5deep tool will reouse system to collect such hashes
- National Software Reference Library (NSRL)
  - ○ NIST project to maintain database of hashes for all "known traceable software applications"
  - ○ Downloadable from http://www.nsrl.nist.gov/
    - – NSRL motto: *you never can have too much overkill*

The Sleuth Kit

TSK contains low level 20 tools to analyze disk images

- – Brain Carrier, www.sleuthit.org
- ≈ Reformat disk image for use by other tools, eg
  - ○ MACTIME: timeline analysis
  - ○ SORTER: hash comparisons, type matching, keyword searching
- – Tools designed to analyze and reformat data at a specific file system layer
  - ○ eg, ifstat displays details of a specific inode

Autopsy

GUI front end for TSK and standard UNIX utilities

- Imports and analyzes images in standard formats
  - NTFS, FAT, FFS, EXT2FS, and EXT3FS file systems
- Organizes and maintains data by a “Case”
- Performs standard forensic functions
  - Creates timeline
  - Recovers deleted data
  - Word searches (files, deleted files, slack space)
  - Recovers context and file details for found keywords
  - Hash comparisons
- HTML output

○ GNU license

DEMO

Open Autopsy, create case for the hacked Windows 2003 server, import image and run tools

1. What should we expect from a check of file hashes against database of “known_good” ?
2. When did the tapering occur?
3. What was tampered with?
4. How do we determine what accounts were involved?
5. How do we determine the vulnerability that was exploited.

Reporting

General principles

- Provide detailed narrative of your activities
  - ○ Tools
  - ○ Protections given to evidence
- Objectively describe observations (ie output from tools)
- Draw conclusions based on evidence and facts related to the systems involved.

Real World

Internal investigation

- Focus on solving the problem & preventing a repeat. High values for:
  - ○ timeliness
  - ○ reliability
  - ☼ Utility

If investigation did not bring about change, then it was [largely] a waste of time & resource

Real World

External Investigation | Adversarial situation

1. Forensic results are just one kind of evidence
2. What the *attorneys* want to argue is what is important
3. Use of the report will require testimony of an expert
   - ○ Demonstrate the reliability of the tools and the applied forensic methods
   - ○ Appropriateness of the conclusions based on the data

Thank You

○ Questions?