Digital Forensics
Lecture 4

Collecting Volatile Data

Additional Reference: **Computer Evidence: Collection & Preservation**, C.L.T. Brown

Current, Relevant Topics

- **Cops follow texting trail**
  - **Sunday, August 27, 2006**
  - Ten minutes before a deadly midnight shooting in a Fair Oaks park, a man sent a text message claiming that he is wanted for murder.
  - Hours before the homicide, a woman sent invitation messages to the man who ended up murdered
  - The woman, Mariya Stepanov, 19, is charged with homicide. The man will be a court witness.

Nationally, text messages are popping up in high-profile cases: murder, rape, trace missing or abducted people,

sacbee.com (Sacramento Metro/Regional News)

This Week's Presentations

0011

1. **Volatile Data**

2. **Tools for Live Collection**

Research Topics Presentation (Due Next Week)

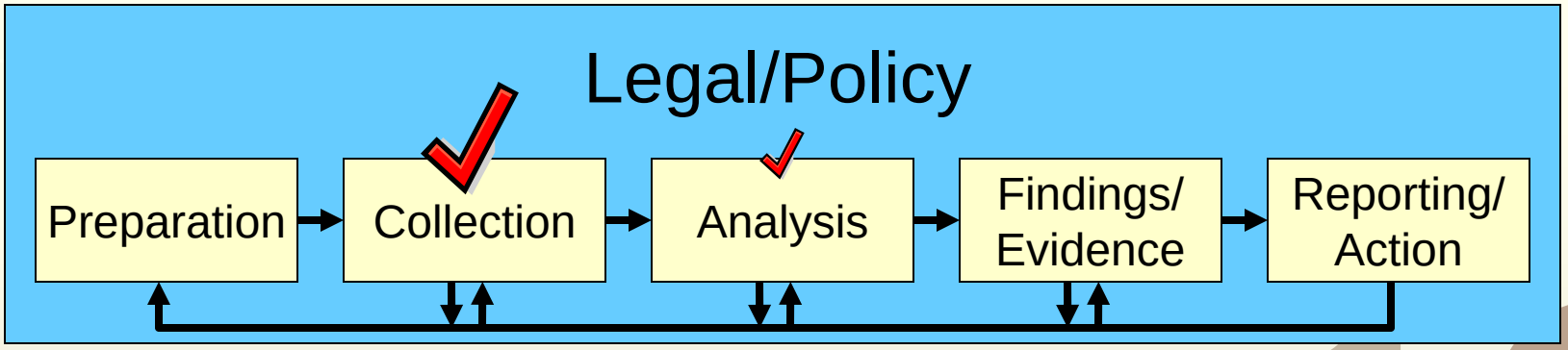
0011

We are counting on you for specifics

Analysis Techniques: keyword searches, timelines, hidden data, ...

- **File Encoding and Detection**
- **Timeline Analysis**
- **Data Mining for Digital Forensics**
- **Encryption and Password Recovery**
- **Steganography Detection**
- **File Extension Renaming and Signaturing**

Lecture Overview



1. Introduction to Volatile Data
2. What to collect
3. How to collect
4. Process
5. Toolkits
6. Security Focus Live Collection
7. Windows Live Collection

Module 1

Introduction to Volatile Data

Volatile Data

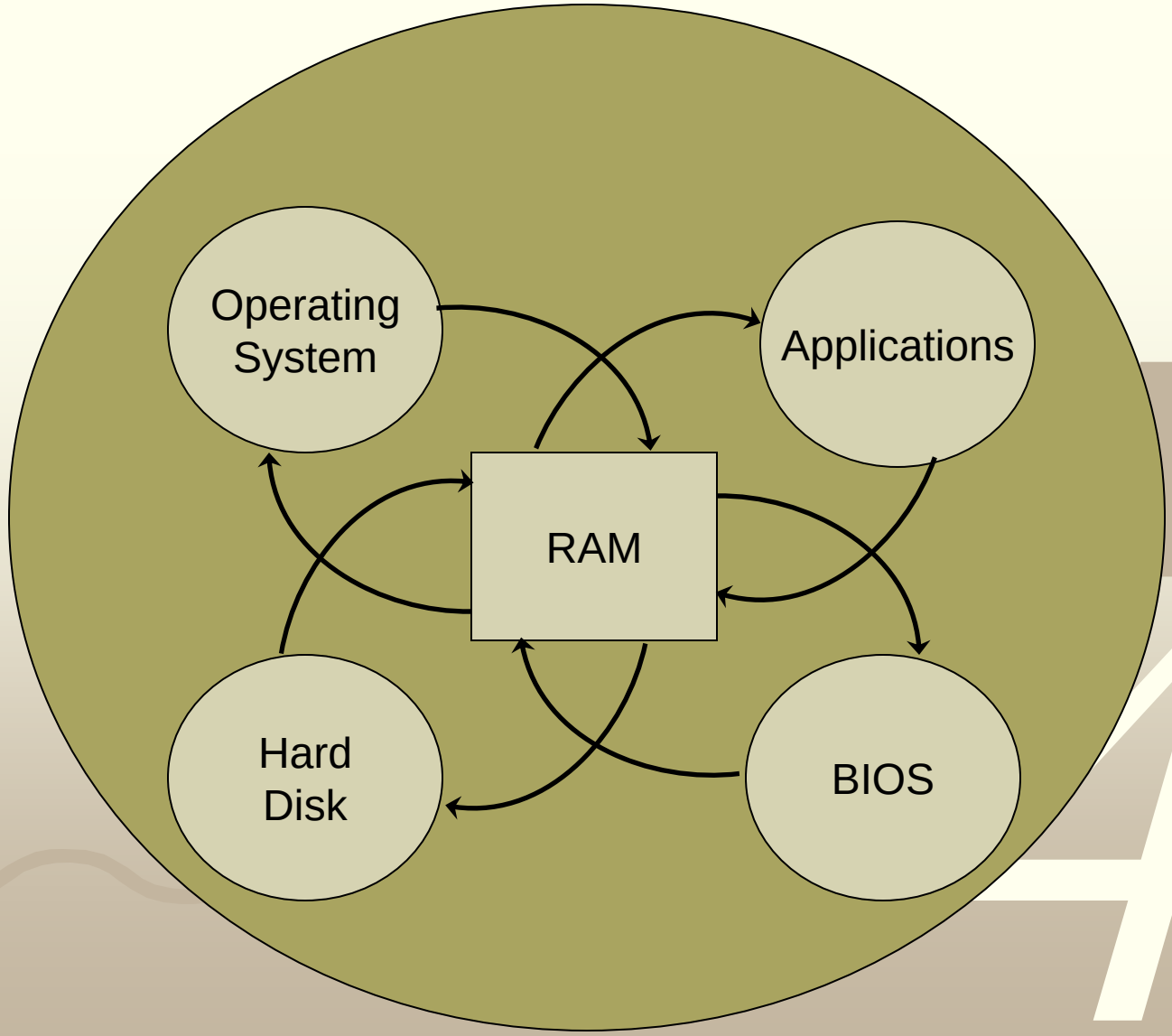
- Data in a state of change.
- Data lost with the loss of power.
- Information or data contained in the active physical memory.
- System Data
  - physical volatile data – lost on loss of power
  - logical memory – may be lost on orderly shutdown

Considerations

- May not be able to shutdown systems without destroying data or causing financial loss.

- If a system is in the process of destroying data, the system needs to have the plug pulled to stop the loss.

Computer System State



Where do we find Volatile Data

- Physical Memory- OS keeps key functions and data here
- Registers - OS keeps key functions and data here
- Virtual Memory in the file system
- Peripheral device memory

Windows Rootkits

- 1st Generation
  - file system rootkits
  - user-mode rootkits
- 2nd Generation
  - volatile memory rootkits – library (dll) rootkits
  - user-mode rootkits, e.g., Hacker Defender
- 3rd Generation
  - device driver rootkits
  - kernel-mode rootkits, e.g., Vanquish, HE4Hook

Detecting kernel-mode rootkits: Network-enabled computer forensics to create bit-stream images of physical memory, e.g., ProDiscover or EnCase Enterprise Edition

Caveats

- By **running any tools** on a live system we load them into memory and create at least one process which can overwrite possible evidence. By creating a new process, the memory management system of the operating system allocates data in main memory and then can overwrite other unallocated data in main memory or in the swap file system.
- The signs of intrusions found in images of main memory can be untrusted, because they could be created by our acquisition tools.

Module 2

What to Collect

Operating Systems

- Code page loaded in memory for execution
  - low level IO functions
    - loaded in physical or logical page memory
    - may be cleared on shutdown
  - may contain passwords, etc.
    - Example: Trillion chat client
      - "pwd=…"
  - may contain hacker handles, group names, etc.
    - Example: Hacker Defender rootkit
      - "hxdef-rk073s.\\.mailslot\hxdef-rkc00…"
  - users may configure a limit to time in memory

Routers & Appliances

- Cisco Router
  - lacks hard drive -> flash memory
    - Internetwork OS & supporting files
  - Dynamic or Synchronous RAM
    - Volatile data: running OS, routing table, statistics, local logs, …
  - Non-Volatile RAM - startup configuration files
  - BootROM - code for power-on self-test, IOS loading, …
  - Router Security Audit Logs
    - allow remote tracking of changes to router configuration
  - Console Port or AUX port
    - support to run a terminal session

Open Problem: few options for collection of evidence

PDAs, Cell Phones, MP3 Players, …

- Storage in flash cards is common
- PDA
  - CPU, RAM, peripheral ports, etc.
  - Memory sticks, Secure Digital cards, …
  - Primary storage for user data is in RAM, which is kept in place by device's power

Caveat: Ensure that power is maintained to protect data.

Incident Response on Live Systems

- What to collect
  - Raw memory
  - Users: successful and failed logons, local & remote
  - Processes: running processes and dependencies
  - Network: IP connection, configuration, route tables, MAC address-resolution cache, …
  - Date & Time: configuration settings
  - Task Management: tasks scheduled

Basic rule: gain the most potential evidence reliably with least intrusion.

Module 3

How to Collect

Raw Versus Processed Memory

Raw

- Less intrusive method
- Information is more complete
- Redirect physical memory dump to external memory via network, USB, or Firewire.

Processed

- An application (from trusted binary CD) issues OS calls via API to extract information from physical memory
- More immediately useful
  - logged on users
  - processes
  - TCP/IP connections
  - Ports
- Incomplete information

Tools

- Regmon
  - what registry keys are being accessed
- Filemon
  - what files are bin accessed
- Tribble
  - hardware expansion card to acquire volatile memory of a live system

www.sysinternals.com

Module 4

Process

Live Analysis

One approach

- Disconnect the system from the network
  - unless you are trying to track an active attack
  - usual case you want to stop any damage or loss of valuable information
- Record
  - Note the time, date, who discovered the problem and how you were made aware of it. From now on every time you do something make a note of the situation describing what actions were taken, what results were found, and when & where it took place.
- Evidence
  - Forensics code (TCT) on a CDROM or other immutable media, ready for action – collect information
- Determine Action
  - Based on the data collected decide what to do…

Actions

- False alarm
  - Do nothing – resume normal operation
- Attacked system
  - Catching attacker is critical
    - set a trap or track the intruder
  - Recovering / protecting the system is critical
    - perform damage control to any serious bleeding wounds
    - secure your system

After the Incident Recovery

- Create a security policy. Document the changes to secure your system as an excellent start to policy.
- Install any and all vendor security patches.
- Turn off all network services that you don't use, use one-time passwords (logdaemon and s/key), encrypted login sessions (ssh), and run security/auditing tools on your system.
- Learn your system better.
- Turn on logging & accounting and look at them!
- Create a baseline: Create backups, run MD5's, save output of a TCT run, etc., and secure them to compare against later.
- Regularly audit or at least examine your systems.

Live Analysis

- Run a network sniffer to capture communication flows to and from a compromised system
  - *tcpdump* raw format to reduce performance issues
- Create a paper copy of our data collection procedure
- Record the results of commands run during data gathering while sending all digital data to a remote host or storing it on external media.

http://www.securityfocus.com/infocus/1769

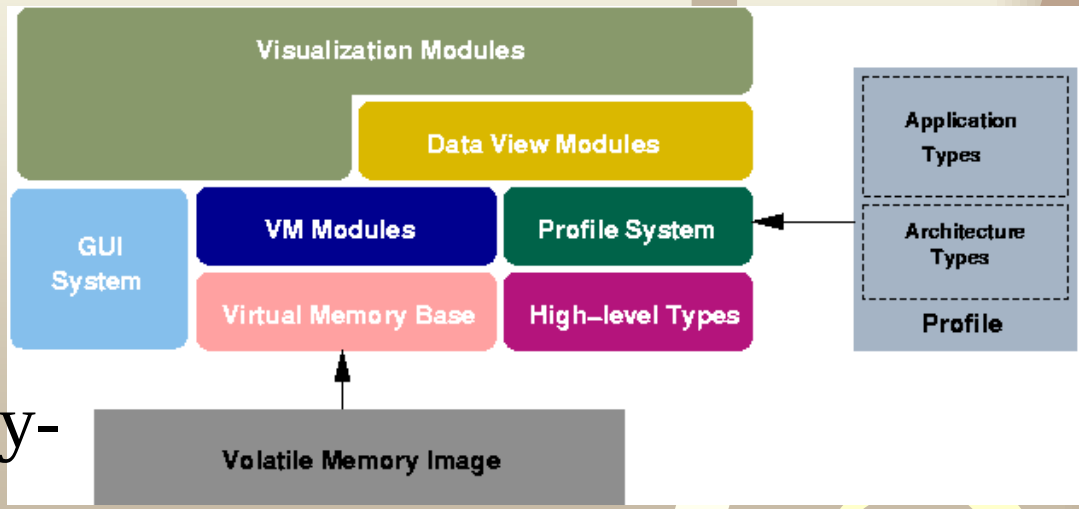
Module 5

Toolkits

http://www.forensics.nl/toolkits

FATKit: The Forensic Analysis ToolKit

- Cross-platform, modular, and extensible digital investigation framework for analyzing volatile system memory
- Automates the extraction and visualization of digital objects found in physical memory
- Linux- and Windows-specific kernel analyses including process/task enumeration, module enumeration, and memory-resident malicious code detection.



State of development in question.

http://www.4tphi.net/fatkit/

Helix Live CD

- Customized distribution of the Knoppix Live Linux CD
- Does NOT touch the host computer in any way and it is forensically sound
- **Windows** functionality to facilitate the capture of live Windows systems' volatile data - runs as a standard windows application
- **Linux** functionality for a bootable, self-contained operating system that can be used for in-depth analysis of "dead" systems.

Free

http://www.e-fense.com/helix

Knoppix-STD 0.1

- Collection of hundreds if not thousands of open source security tools
- Live Linux Distro
- Turn it into a firewall, a web server, an IDS box, a honeypot.
- Use it to do data recovery on an **dead** or locked computer, perform a vulnerability assessment, a penetration test, perform an autopsy on a compromised machine, test your incident response team.

http://s-t-d.org/faq.html

Free

LiveWire Investigator

- LiveWire Investigator captures relevant data – including running state – while the system being investigated continues to operate
- Extensive array of data acquisition options and analytical tools
- Automates the logging and reporting of all investigative actions
- Capture and record running state
  - Volatile Memory Snapshot
  - Live Registry Examination
  - System Log
- Collect key information on running programs, network connections, and data transmissions
  - IP, NetBIOS, Routing table acquisition
  - Running processes

$8995.00

http://www.wetstonetech.com/catalog/item/1104418/2347979.htm

Penguin Sleuth

- Virtual computer forensics and security platform
- Originally modified the Knoppix distribution to make it more forensic friendly
- Knoppix provides a method of looking at the computer, without altering the evidence.
- Detecting files that have not been deleted
- Validated for live preview of EXT23, FAT32, and NTFS partitions.4” . Not validated for live preview of EXT3 or reiserfs partitions
- Live previews of computers: chkrootkit, tcpdump, and many other live network analysis tools

http://www.penguinsleuth.org/

http://www.linux-forensics.com

Free

The Coroner's Toolkit (TCT)

- Primarily designed for Unix systems, but it can do some data collection & analysis on non-Unix disks/media.
- Tools
  - grave-robber (data capturing tool)
  - the C tools (*ils*, *icat*, *pcat*, *file*, etc.)
  - *unrm* & *lazarus* (collection & analysis of data on deleted files)
  - *mactime* (analyzes the mtime file)
  - *findkey* tool that recovers cryptographic keys from a running process or from files.

http://www.porcupine.org/forensics/tct.html#features

Free

EnCase Enterprise Edition

- Network-enabled, multi-platform enterprise investigation solution.
- Preserves volatile and static data on servers and workstations anywhere on the network, without disrupting operations.
- Extensive capability
  - Securely investigate/analyze machines over the LAN/WAN from a central location
  - Audit machines for compromise by zero-day attacks
  - Identify and remediate Windows-based kernel rootkits

www.guidancesoftware.com

expensive, but must get a quote

ProDiscover

- Computer Forensic Tool for Law Enforcement
- Find all the data on a computer disk while protecting evidence and creating evidentiary quality reports for use in legal proceedings

- Examines disk – does not appear to do live analysis.

$7995.00

www.techpathways.com

Tradeoffs

- Windows-based operating systems
  - beware that even moving the mouse accesses dynamic registry hives

- Displacing a few bits of volatile memory may be worth identifying a password cached in memory…

Module 6

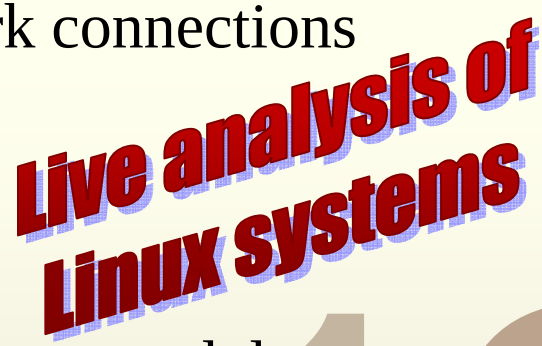
Security Focus Linux Live Collection

http://www.securityfocus.com/infocus/1769

Building a Live Collection Disk

Security Focus Live Collection Disk and Process

- *nc* – read and write data across network connections
- *dd* – copy and convert a file
- *datecat* – print date information
- *pcat* – print process information
- *Hunter* – print process info for suspicious modules
- *insmod* – install loadable module in the running kernel
- *netstat* – list currently active network connections
- *arp* – manipulate the kernel's ARP cache
- *route* – manipulate the kernel's IP routing tables
- *dmesg* – examine or control the kernel ring buffer



http://www.securityfocus.com/infocus/1769

Basic idea

- Do the least intrusive volatile data collection first then proceed to more intrusive and less volatile data collection

Linux Live Collection Process

- Photograph what is on the screen, better yet videotape your collection!
- Mount external media into the compromised system
  - we have no choice but to use the untrusted command at this point – this will be the location for our trusted commands (if we have to unmount a disk use a trusted *unmount* on a floppy)
  - this will modify atime in /etc/ld.so.cache, /lib/tls/libc.so.6, /usr/lib/locale/locale-archive, /etc/fstab, /etc/mtab*, /dev/cdrom, /bin/mount
  - (*also ctime & mtime; avoided by using –n)

Process Continued

- All results generated by trusted commands have to be sent to the remote host.
  - Use netcat and the pipe method:
    - run a trusted shell
      - (compromised)# */mnt/cdrom/bash*
    - open TCP port on the remote host
      - (remote host)# *nc -l -p 8888 > date_compromised*
    - send the date of the compromise to record
      - (compromised host)# */mnt/cdrom/date* | */mnt/cdrom/nc* 192.168.1.100 8888 -w 3
    - document the hash of the collected data
      - (remote host)# *md5sum date_compromised > date_compromised.md5*

Process Continued

- Collect current date
  - (remote)# *nc* -l -p port > date_compromised
  - (compromised)# ***/mnt/cdrom/date* -u** | */mnt/cdrom/nc* (remote) port
  - (remote)# *md5sum* date_compromised > date_compromised.md5

Note: in our example (remote) port is 192.168.1.100 8888

Process Continued

- Collect Cache Tables as they are extremely volatile - here arp and routing tables are collected
  - Mac address cache table:
    - (remote)# *nc* -l -p port > arp_compromised
    - (compromised)# ***/mnt/cdrom/arp -an*** | /mnt/cdrom/nc *(remote) port*
    - (remote)# *md5sum arp_compromised > arp_compromised.md5*
  - Kernel route cache table:
    - (remote)# nc -l -p *port* > route_compromised
    - (compromised) # ***/mnt/cdrom/route -Cn*** | *(remote) port*

Process Continued

- Current, pending connections and open TCP/UDP ports
  - (remote)*#nc -l -p port > connections_compromised*
  - (compromised)# ***/mnt/cdrom/netstat -an*** | */mnt/cdrom/nc (remote) port*
  - (remote)# *md5sum connections_compromised > connections_compromised.md5*
  - an easy method of detecting a rootkit, loaded into kernel memory, is when one of its tasks is hiding an open port.

Process Continued

- Physical memory image
  - access physical memory directly by copying the /dev/mem device or by copying the kcore file mounted in the /proc directory
  - kcore is in the ELF core format, so it can be debugged later by the gdb tool.
  - In page tables we can find the order of pages (4 KB in Intel processors per page) written to the physical memory.
    - (remote)# *nc -l -p port > kcore_compromised*
    - (compromised)# ***/mnt/cdrom/dd < /proc/kcore*** | */mnt/cdrom/nc* (remote) port
    - (remote)# *md5sum kcore_compromised > kcore_compromised.md5*
  - This copies both allocated and unallocated data

Process Continued

- List modules loaded in kernel memory
  - (remote)# *nc -l -p port > lkms_compromised*
  - (compromised)#***/mnt/cdrom/cat /proc/modules*** | */mnt/cdrom/nc (remote) port*
  - (remote)# *nc* -l -p port > lkms_compromised.md5
  - (compromised)# */mnt/cdrom/md5sum /proc/modules* | */mnt/cdrom/nc (remote) port*

Process Continued

- Some malicious modules cannot be listed at all.
  - (compromised)#***/mnt/cdrom/insmod -f /mnt/cdrom/hunter.o***
- The "-f switch", forces the loading of the hunter.o due to version mismatch with kernel (If we know which kernel version is on the compromised machine we can download the proper source code from www.kernel.org)
  - (remote)# *nc -l -p port > modules_hunter_compromised*
  - (compromised)# ***/mnt/cdrom/cat /proc/showmodules && /mnt/cdrom/dmesg*** | */mnt/cdrom/nc (remote) port*
  - (remote)# *md5sum*

Process Continued

- Copy the **symbols** exported by kernel modules. By analyzing the ksyms file we can detect the presence of an intruder in the system.
  - (remote)*#nc -l -p port > ksyms_compromised*
  - (compromised)#***/mnt/cdrom/cat /proc/ksyms*** | */mnt/cdrom/nc (remote) port*
  - (remote)*# nc -l -p port > ksyms_compromised.md5*
  - (compromised)*#/mnt/cdrom/md5sum /proc/ksyms* | */mnt/cdrom/nc (remote) port*

Process Continued

- List of Active Processes
  - When we don't detect any LKM based rootkits in memory.
    - (remote)# *nc -l -p port > lsof_compromised*
    - (compromised)# ***lmnt/cdrom/lsof -n -P -l*** | */mnt/cdrom/nc (remote) port*
    - (remote)# *md5sum lsof_compromised > lsof_compromised.md5*
  - Analyze the result from the *lsof* tool. If any of the active processes are suspicious, copy them.

Process Continued

- Examples of suspicious processes:
  - A process is listening on an atypical TCP/UDP port or open raw socket;
  - A process has an active connection with a remote host;
  - A program that was previously run has since been deleted;
  - A file, opened by a process, is deleted (for instance: a log file);
  - A strange process name;
  - A process was initiated by a user that does not exist, or by an unprivileged user.

Process Continued

- Collecting Suspicious Processes
  - (remote)# *nc -l -p port > proc_id_compromised*
  - (compromised)# ***/mnt/cdrom/pcat proc_id*** *| /mnt/cdrom/nc (remote) port*
  - (remote)# *md5sum proc_ip_compromised > proc_ip_compromised.md5*

Process Continued

Useful information about the compromised host

Command	
/*mnt/cdrom/cat* /proc/version	OS Version
/*mnt/cdrom/cat* /proc/sys/kernel/name	Host Name
/*mnt/cdrom/cat* /proc/sys/kernel/domainame	Domain Name
/*mnt/cdrom/cat* /proc/cpuinfo	Hardware info
/*mnt/cdrom/cat* /proc/swaps	Swap partitions
mnt/cdrom/cat /proc/partitions	Local file systems
/*mnt/cdrom/cat* /proc/self/mounts	Mounted file systems
mnt/cdrom/cat /proc/uptime	Uptime

Process Continued

- Record the current time.
  - (remote)# *nc -l -p port > end_time*
  - (compromised)# ***/mnt/cdrom/date*** | */mnt/cdrom/nc (remote) port*

- Switch off the compromised system: pull the power cable from the system or UPS device.

Process Continued

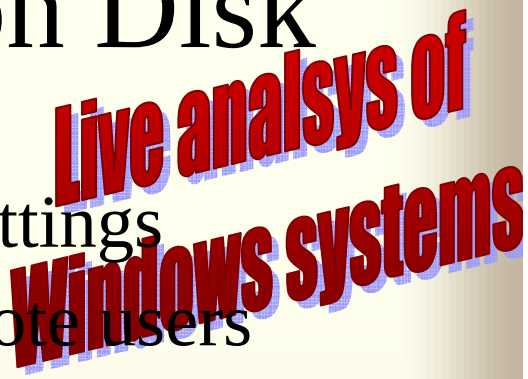
- Now ready for collection of standard “dead” system image.
- Also ready to analyze the “live” data collected here.

Module 7

Windows Live Collection

Computer Evidence: Collection & Preservation; C.L.T. Brown

Building a Live Collection Disk



- NET ACCOUNTS – account policy settings
- NET FILE – display open files by remote users
- NET SESSION – display remote connections
- NET SHARE – display local directory shares accessible from network
- NET START – display services and their status
- NET USE – display remote network shares to which system is currently connected
- NET USER – display all user accounts
- NET VIEW – display computers in local domain

Building a Live Collection Disk

- *Route Print* : display local system's current route tables
- *ARP –a* : display current MAC address to IP address mapping
- *NETSTAT –anr* : display connections and listening ports
- *NBSTAT –c* : display current NetBIOS name cache with remote machine names and IP addresses
- *AT* : Display scheduled command scheduler operations

Building a Live Collection Disk

- *PSList* : list detailed information about processes
- *PSInfo* : list information about a system
- *PSLoggedon* : List users logged on locally and via resource sharing

- *Fport* : show the application associated with an open TCP/IP port
- *Ntlast* : extract security log information

Building a Live Collection Disk

- dd.exe
- md5lib.dll
- md5sum.exe
- Volume_dump.exe
- wipe.exe
- zlibU.dll
- nc.exe
- getopt.dll

- Windows versions of utilities typically used on Linux systems.

Windows Live Collection Process

- Essentially, follow the process used for Linux, but substitute the Windows tools just listed.

- Create trusted batch files and put tools and batch files on write protected media.

Questions?

After all, you are an investigator