

# Forensic Computer Investigations

Steve Romig  
May, 2010

# Definitions & Principles

- What is “Forensic Computer Investigation”?
  - Forensic means “pertaining to the law”
  - We have forensic anthropology, ballistics, genetics, chemistry, liquid splatter analysis, dentistry...
- Good general introduction: “Criminalistics”, by Richard Saferstein

# Why Bother?

- Academic misconduct
- Policy/human resources issues
- Criminal incidents
- Civil incidents
- These same techniques are useful for general investigations on computers
  - The system crashed, why?
  - We were compromised, how?

# Why Bother?

- Some questions to ask:
  - How did they break in?
  - What damage was done?
  - Who did it?
  - Who else did they hit?
- We do it in a “forensically sound way” to:
  - Meet legal requirements
  - Reduce liability
  - Preserve evidence

# The Four Steps

- Good definition:
  - "Process of identifying, preserving, analyzing and presenting digital evidence in a manner that is legally acceptable in any legal proceedings (i.e. a court of law)."
  - Rodney McKennish: "1998 Donald Mackay Churchill Fellowship to Study Overseas Developments in Forensic Computing" (Australia)

# The Four Steps

- Identify the evidence
  - Must identify the type of information that is available
  - Determine how to best retrieve it
  - Disk images, memory dumps, process listings, log files, network traffic logs, etc.
  - We might need to prioritize the evidence, based on what questions we are trying to answer or what we expect to find.

# The Four Steps

- Preserve the evidence
  - With the least amount of change possible
  - You must be able to account for any changes
  - How can you show that what you have now is IDENTICAL to what you had way back then?

# The Four Steps

- Analyze the evidence
  - Extract, process, interpret
  - Extract - evidence collection may produce binary 'gunk' that isn't human readable
  - Process - make it humanly readable
  - Interpret - requires a deeper understanding of how things fit together
- Your analysis should be repeatable

# The Four Steps

- Present the evidence
  - To law enforcement, attorneys, in court, etc.
  - Acceptance will depend on
    - Manner of presentation (did you make it understandable, convincing?)
    - The qualifications of the presenter
    - The credibility of the processes used to preserve and analyze the evidence
    - Credibility enhanced if you can duplicate the process
  - This is especially important when presenting evidence in court

# Investigation Workflow

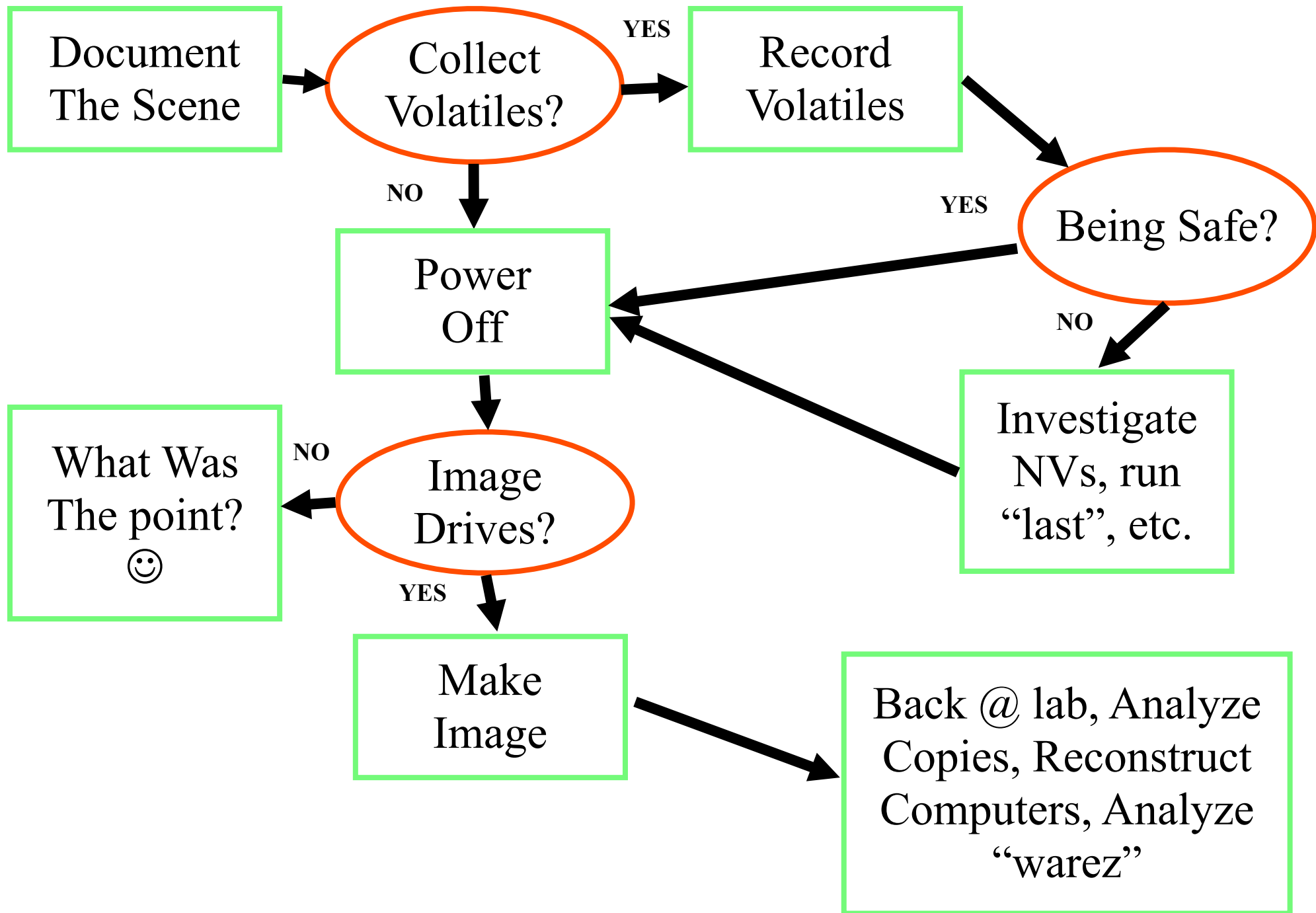
- Collect and analyze evidence to form one or more chronological sequences of events that fit the evidence
- We can't always be conclusive!
  - “The butler did it”
  - “Either the butler did it or he picked up the knife after the murder”
- It is a feedback loop
  - Analysis leads to more evidence which feeds analysis...

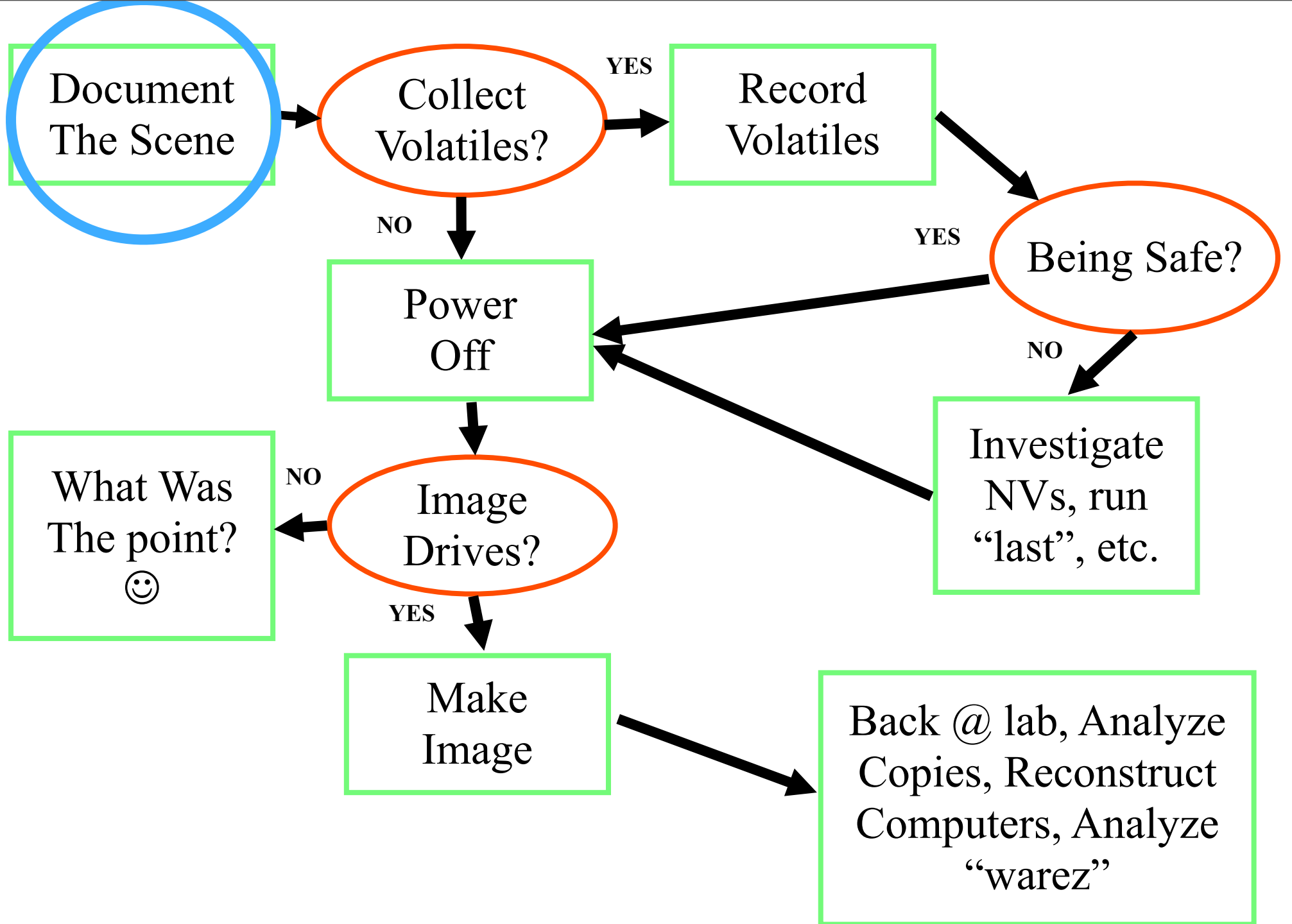
# Five Points to Consider

- **Admissibility**
  - Conform to legal requirements (“rules of evidence”)
- **Authenticity**
  - Relevant to the case at hand
- **Completeness**
  - Complete logs are better than extracts from the logs
- **Reliability**
  - Collected and handled appropriately
- **Believability**
  - Understandable and convincing

# Legal Issues

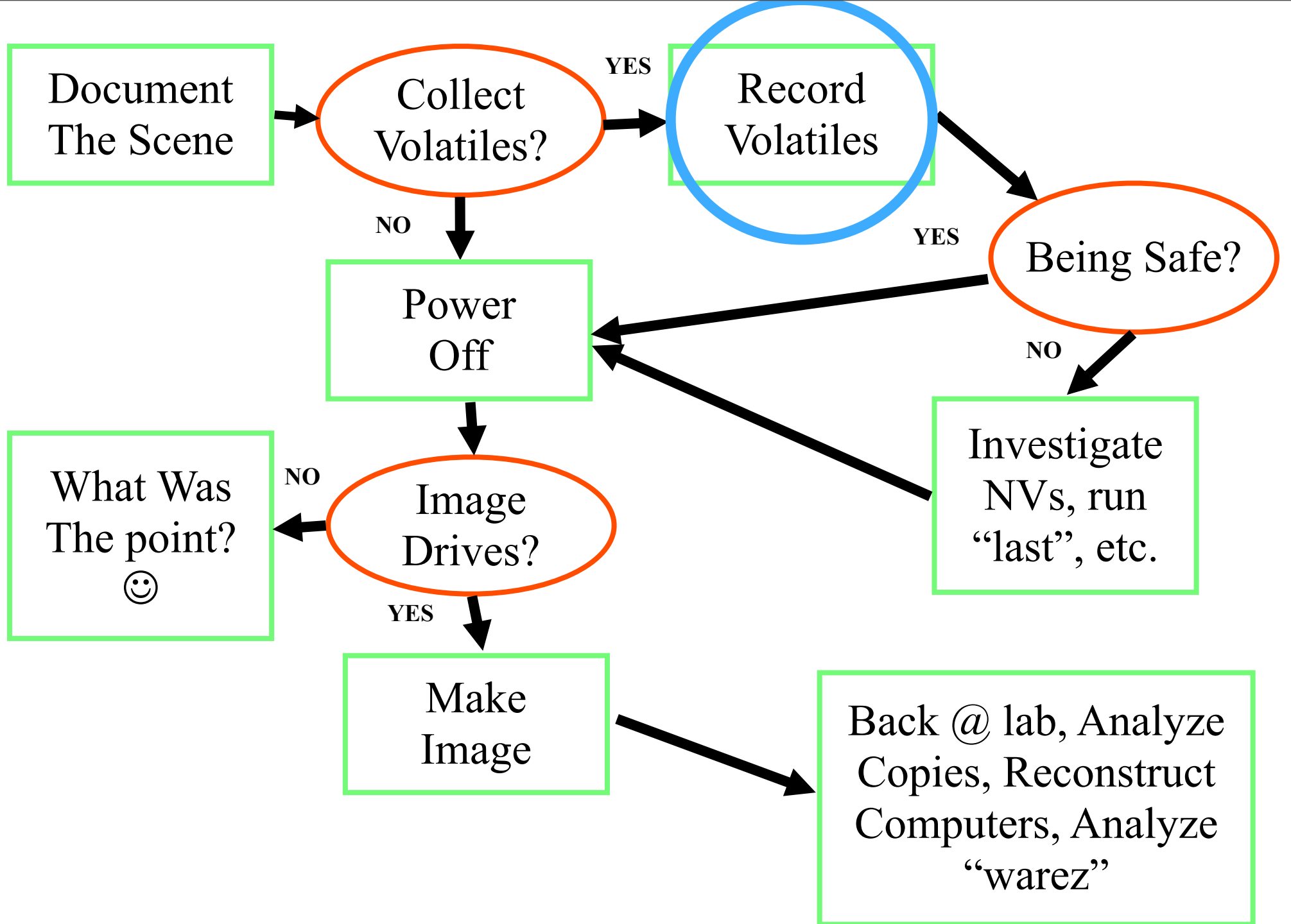
- Best Evidence
- Hearsay
- The Frye and Daubert Tests
- Chain of Custody
- Exculpatory Evidence
- Fruit of the Poisonous Tree
- Acting Under Color of Law





# Document the Scene

- Map the room(s)
- Take pictures
- Label everything
  - Permanent, or removable sticky notes (NOT post-it notes – they fall off)
  - Unique “tag” – 315-1-2 (room 315, computer 1, disk 2)
- Catalogue everything



# Collect Volatile Evidence

- “Volatile evidence” is evidence that will disappear soon, such as information about active network connections, or the current contents of volatile memory.
- Contrast this with the contents of a disk or tape.

# Collect Volatile Evidence

- From Farmer & Venema – <http://www.porcupine.org>
  - Registers, peripheral memory, caches...
  - Memory (virtual, physical)
  - Network state
  - Running processes/services
  - Loaded kernel modules/dlls/drivers
  - Network shares
  - Mounted file systems

# Collecting Volatile Evidence

- What you do on the system will affect the remaining evidence
  - Running 'ps' will overwrite parts of memory
  - Your shell may overwrite the history file
  - You may affect file access times
  - There's always the risk of trojans! (e.g. gcore)

# Collecting Volatile Evidence

- Rootkits
  - Everything you know about a system is given to you through the software you use (the applications, the libraries, the operating system)
  - A rootkit is software that subverts the system to hide processes, files, network connections and so on
  - These often contain back doors, which give the intruder easy return access

# Collecting Volatile Evidence

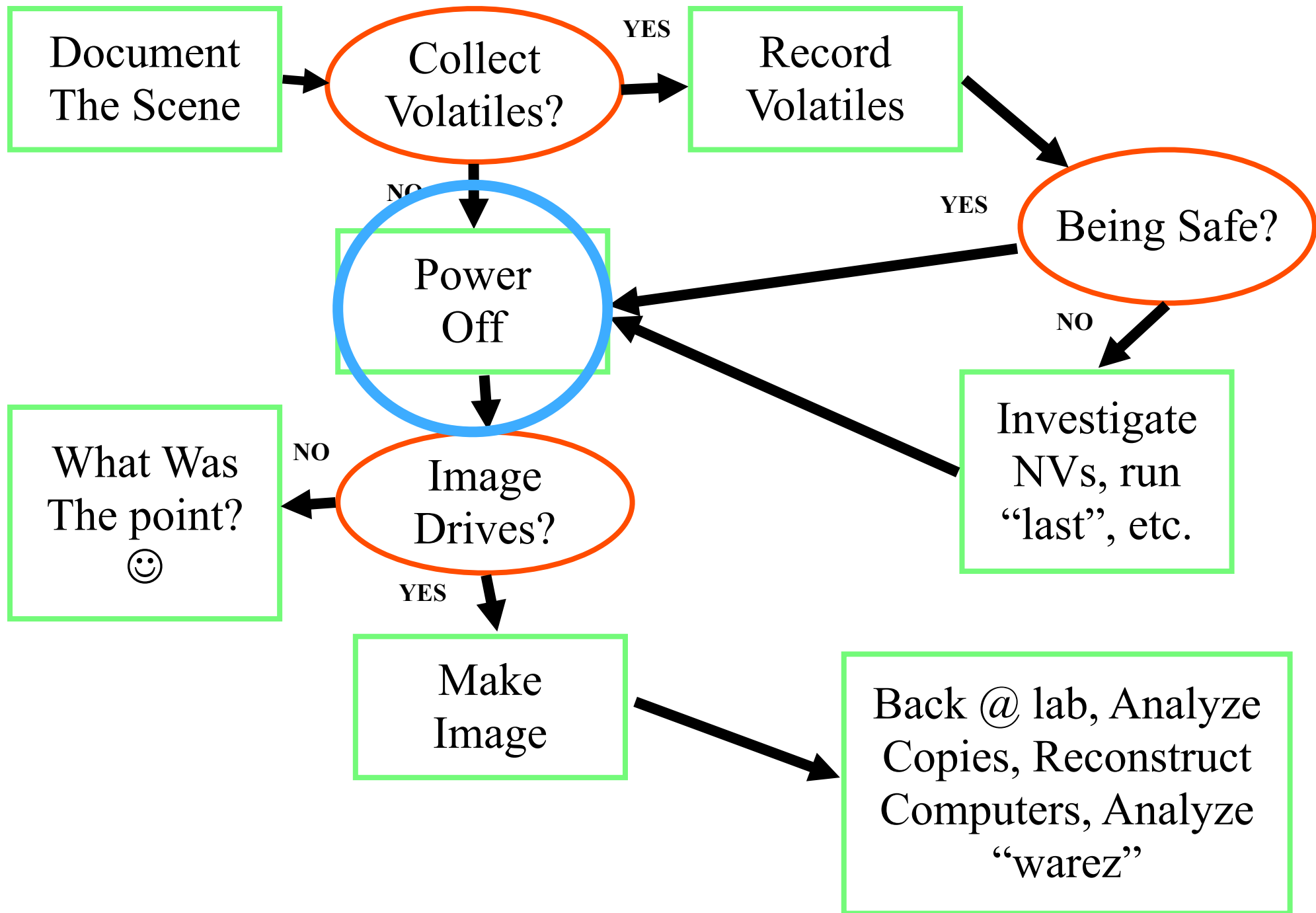
- You need to use known, safe tools to examine a system
  - Statically linked
  - Or include your own libraries
  - Mount from floppy or CD, through net, or download through net
- This won't help with kernel rootkits

# Collecting Volatile Evidence

- Your toolkit might include:
  - sysinternal's filemon, regmon, process explorer, tcpview, autoruns, rootkit-revealer, dumpevt, dumpreg...
  - f-secure's blacklight
  - icesword
  - microsoft's windows defender

# Collecting Volatile Evidence

- If you are collecting volatiles
  - Download/mount your tools (net, floppy, cd, flash)
  - Copy memory, swap, /tmp, pagefile.sys...
  - Get info about network state (connections, promiscuous interfaces)
  - Get info about running processes
  - Write results to flash or across the network: never to the local hard drive



# Turning a computer off...

- When you examine a computer, should you:
  - Turn it off? Use the switch vs. battery/cord?
  - CTRL-ALT-DELETE, L1-A?
  - Reboot?
  - Unplug it from the net?
  - Filter it at the router?
  - Leave it running and examine it quickly?

# Three Fingered Salute

- CTRL-ALT-Delete, L1-A (Suns), etc.
  - Can be caught, redirected to destruct routines
  - No real advantage to doing this (that I can think of; you might as well just power off).

# Shutdown

- Shutdown/halt/sync would leave file systems clean
  - But these routines might be rigged for destruction
- Don't reboot!
  - Worse than doing a shutdown!
  - Wiping /tmp on reboot (if it isn't a ram-disk)
  - Is it rigged to restart "bad stuff" (backdoors, destructive things) at reboot? Or later, through cron?

# Unplug from the Network

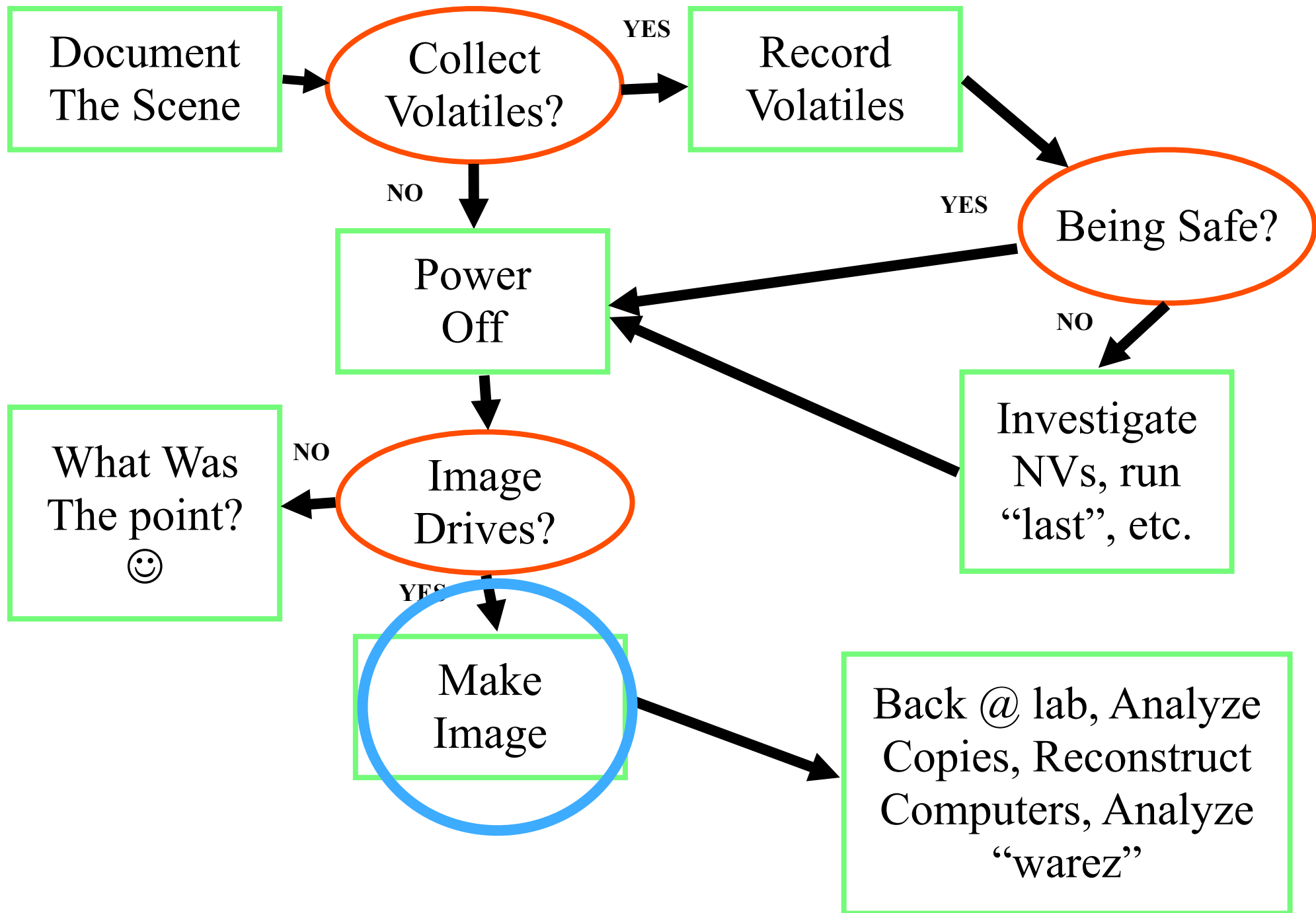
- If you unplug from the network or filter it...
  - What about "dead man switches" that detect when they're off the net and wipe evidence?
  - Marcus Ranum wrote about this in the CSI Alert, September 1999, #198

# Leave it Running

- Without unplugging from the network
  - Until you power it off
- This is probably safe in the short term
  - Risk increases with time, though
  - They might use it to do nasty business - liability?
  - They might wipe evidence, especially if they see you poking around

# Power Off

- When you turn it off...
  - You lose volatile evidence: processes, network connections, mounted network file systems, contents of memory...
  - This is critical evidence in many cases: crackers increasingly store tools, logs on remotely mounted file systems
  - On the other hand, if you investigate on running system, you risk modifying the system (especially the disk)

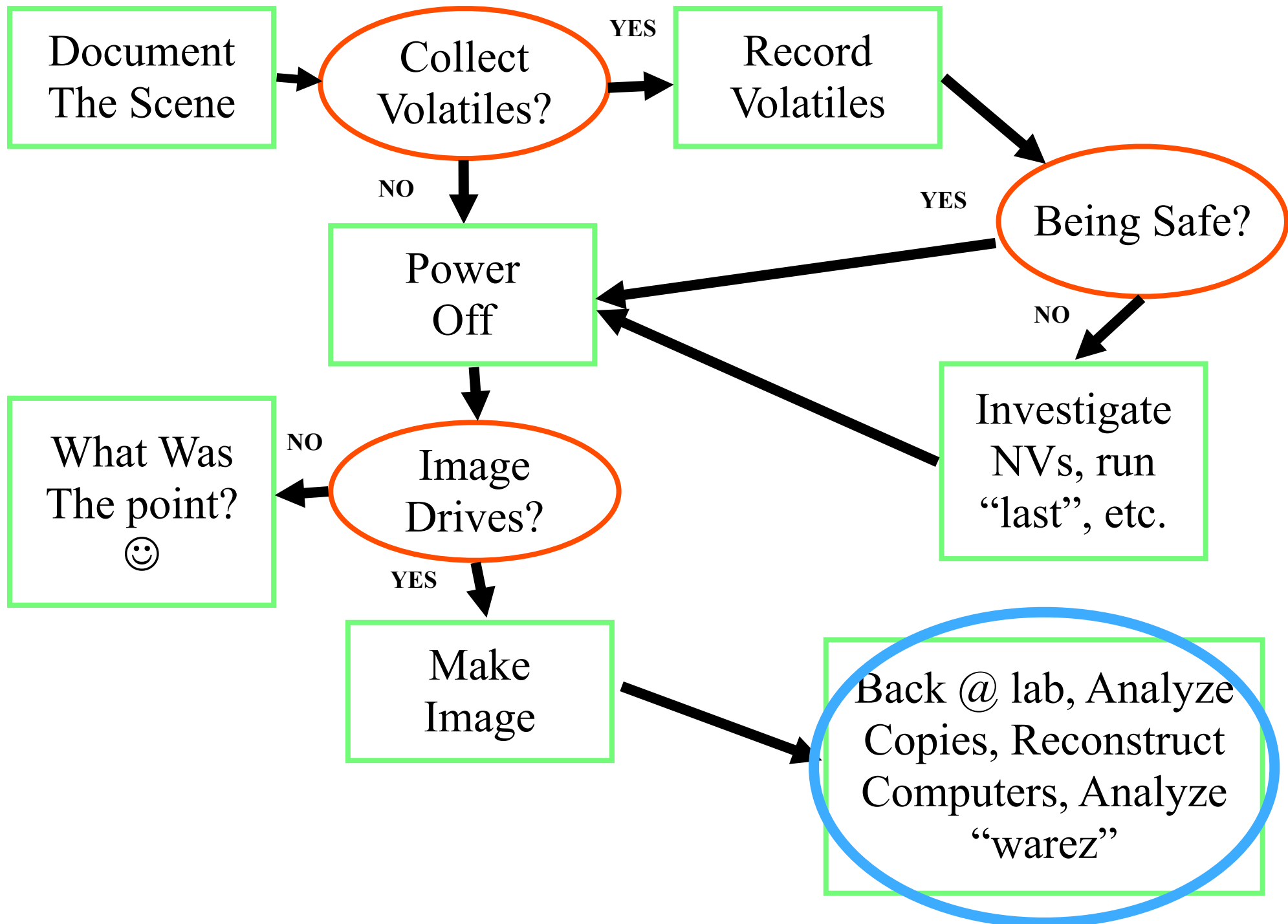


# Imaging Disks

- Get partition, RAID, logical volume management configuration
- Make copies of the hard drives (or RAIDs, or partitions, or...)
- Calculate and compare hashes (MD5 and/or SHA-1)
- Document and witness the copy/verification!
- Reconstruct RAIDs, carve our logical volumes, etc.

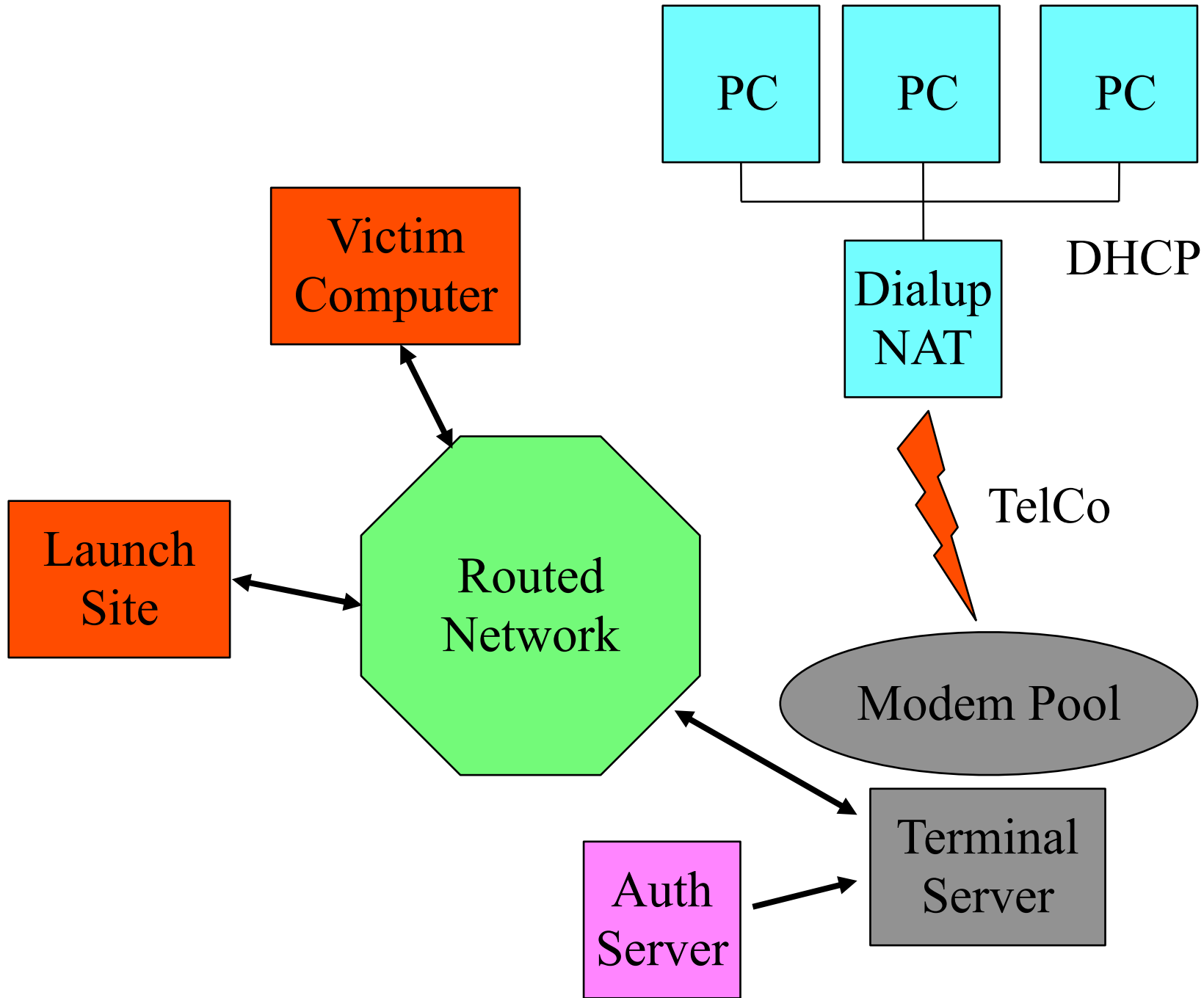
# Imaging Disks

- Common tools include:
  - Helix, Knoppix live CDs
  - SMART (Linux live CD) from ASR
  - Forensic ToolKit (FTK) from Access Data
  - EnCase from Guidance Software
  - FTK Imager
  - Raid Reconstructor from Runtime Software
  - Unix dd, md5sum



# We need to know:

- Where the evidence is
- What the evidence means
- How to put it together



# Where the Evidence is

- home system
- phone system
- modem pool
- networks
- victim computers
- think about the components
- ask questions, get expert advice

# What the Evidence Means

- This requires a deeper understanding
  - How evidence is created
  - Where it might be missing
  - Or wrong
- Get an expert, ask questions

# What the Evidence means

- A romig login entry in a UNIX wtmp file means...
  - Someone used the romig account to login
  - Or inserted a fake entry
  - NOT necessarily that Steve Romig logged in
- A DHCP lease means...
  - A computer was assigned the lease
  - NOT that that computer was the one using that IP address during the lease time

# Importance of knowing

- Where the logs might be wrong
  - Syslog, NetFlow exports are sent via UDP
  - Authentication logs from parallel authentication servers
  - NetFlow logs and asymmetric routes
  - Spoofed IP addresses
  - Writable logs (wtmp, utmp on old UNIX systems)
  - Logs modified by the cracker

# Correlating Logs

- You can build stronger case if you can show multiple sources that are in agreement
- Relating log entries to each other
  - Matching log entries by value - e.g. IP address
  - Matching entries by time

# Time Related Issues

- We often use timestamps to correlate entries from different logs on different systems
- Problems include:
  - time synchronization
  - time zone
  - event lag
  - chronological order of events
  - event bounding

# Time Synchronization

- We can sometimes infer clock offset from the logs
  - shell history on computer A shows telnet B at T1, tcp wrapper on computer B shows telnet from A at T2
  - offset is (probably  $T2 - T1$ )
- We can't always do this - not enough info, event lag, etc.

# Time Zone

- You can't compare apples to oranges
- Send, request time zone for all logs
- I like GMT offsets
- Make sure you do the math right

# Event Lag

- Event lag is the difference in time between related events in different types of logs
  - connect from computer A to computer B using telnet and login
  - NetFlow log shows telnet starting at 13:05:12
  - TCP wrapper on computer B shows telnet at 13:05:12
  - wtmp shows actual login at 13:05:58
- Lag can be very variable

# Event Lag

- We can use session start time, duration to eliminate some sessions
  - looking for dialup sessions in phone trace that "match" a login session on the modem pool that started at 2:03:22 and lasted 00:10:05
  - sessions that start wayyyy before or after 2:03:22 probably don't match
  - sessions that are short than 00:10:05 don't match
  - sessions too much longer than 00:10:05 probably don't match

# Event Lag

- Session ending time can sometimes be used to match more accurately than starting time
  - hang-up modem, terminal server terminates login session for you - short lag
  - logout of UNIX, telnet session ends - short lag

# Chronological Order of Events

- Some logs are created in chronological order by the ending time of the session
  - process accounting records on Unix
  - Cisco NetFlow logs
  - TACACS+ session summary entries

# Chronological Order of Events

- This can be very confusing
  - look through flow log, see traffic from computer, but not telnet traffic to computer - might not appear until 30 minutes later in the log
  - look through process accounting logs, see sub-processes, but not shell process
- We often need to reorder by the starting time of the session

# Process Accounting Log

ttyp1 romig	12:32:28	00:00:07	ls
ttyp1 romig	12:33:02	00:00:05	cat
ttyp1 romig	12:33:45	00:00:03	egrep
ttyp1 romig	12:33:45	00:00:04	awk
ttyp1 romig	12:33:45	00:00:04	sh
...			
ttyp1 romig	12:30:12	00:10:02	sh

# Event Bounding

- We can use start, end times of one session to "bound" portions of other logs to focus our search for useful information
  - for instance, modem pool auth log shows session from T1 to T2
  - probably not going to find flow logs for the corresponding IP address of interest outside of that session
  - this is obvious

# Event Bounding

- It is not so obvious that we can't always do this
  - easy to leave processes running after your login session on Unix
  - then there's at, cron, procmail and so on
  - these will leave traces long after the modem pool session

# Merging Logs

- Sometimes log entries are spread all over the place
  - multiple parallel authentication servers
  - multiple SMTP front ends
  - multiple routers with asymmetric routing
- Need to merge logs from multiple sources
- Sort into chronological order

# Reliability

- Logs vary in reliability
- How are the logs protected?
  - Some wtmp, utmp are world writable
  - Shell history are writable by their owners
- Depends on the integrity of software that creates log entries
  - Crackers replace these with versions that don't log, or which log false entries - rootkit

# Reliability

- Is subject to the security of transmission over the network
  - syslog, NetFlow both use UDP
  - subject to data loss
  - subject to possible spoofing
- Guard against problems by correlating from as many sources as possible

# Reliability

- We will need to adjust theories to account for anomalies
  - see telnet session to computer, but there's no login session
  - this might indicate rootkit installation
  - doesn't call into question validity of the theory that someone broke into the system - supports it

# IP Address and Host Name Problems

- IP addresses can be spoofed
  - need to recognize cases where this is likely/unlikely
  - common in flooding
  - uncommon in telnet
- Domain stealing, cache poisoning, etc
  - IP address is "better" than the name it resolves to
  - really want to log both
  - if you have to choose one, choose the imp address

# Recognize What's Missing

- Sometimes the stuff that's missing is what's interesting
  - see long telnet in NetFlow to target
  - but there's no login session
  - raises suspicion that there's a rootkit
- We found a ... directory but it doesn't contain anything
  - might be empty
  - might be a rootkit

# What's Missing

- Flow logs shows traffic to TCP/31337
  - but you can't find a process listening on that port
  - there might be a rootkit

# Overview of a Recent Case

- We imaged the physical disk drives
- We “carved” the disks into logical disks used for each RAID
- We reconstructed the RAID as a disk image
- We examined these under EnCase, which allows us to see the partition/volume structure and file system contents

# Overview of a Recent Case

- We extracted file system timestamps, the Internet Explorer history, the Registry contents (with modification times), the IIS logs, all other logs named \*.log, and the event logs
- We converted these to a common format
- We combined and sorted these chronologically, and then started our analysis

# Overview of a Recent Case

- As we identified times when “interesting” activity took place, we would go back to the system image in EnCase and extract the contents of other files, like the malware that was installed.
- We analyzed the malware to try to determine what it was and what it was capable of, how it got installed, files created/read, registry changes, etc.
  - Norman sandbox, Virustotal and Sunbelt Sandbox are useful resources

# Useful Tools

- We use Guidance Software's EnCase, a commercial product (<http://guidancesoftware.com>)
- Sleuthkit & Autopsy - open source alternative (<http://www.sleuthkit.org>)
- Volatility Framework - open source tools for memory forensics (<https://www.volatilesystems.com/default/volatility>)

# More Useful Tools

- Microsoft's Sysinternals tools - autoruns, rootkit revealer, process monitor/explorer, tcpview, regmon, filemon and etc (<http://technet.microsoft.com/en-us/sysinternals/default.aspx>)