



Anti-Forensics and Anti-Anti-Forensics

by Michael Perkin

(But not Anti-Anti-Anti-Forensics)

(...or Uncle-Forensics...)

Outline

- Techniques that can complicate digital-forensic examinations
- Methodologies to mitigate these techniques
- Other digital complications

This talk will deal with a variety of complications that can arise in a digital investigation

Michael Perklin

- Digital Forensic Examiner
- Corporate Investigator
- Computer Programmer
- eDiscovery Consultant

- Basically - A computer geek + legal support hybrid

Techniques in this talk...

- Most of these techniques are NOT sophisticated
- Each one can easily be defeated by an investigator
- The goal of these techniques is to add man-hours/\$\$\$
- High costs increase chances of settlement

Wiping a hard drive, or using steganography will not be discussed because they've been around for decades

Typical Methodologies:

- Copy First, Ask Questions Later
 - Typically Law Enforcement
- Assess relevance first, copy relevant
 - All types of investigators
- Remote analysis of live system,
copy targeted evidence only.
 - Enterprise, Private if they have help



Methodology #1 is typically used by police

Methodology #2 is used by all types of digital investigators

Methodology #3 is typically used by Enterprise companies on their employees

“Assess Relevance” method typically searches an HDD for one of a few specific keywords. If found, the HDD is imaged for further analysis.

Typical Workflow



Create Working Copy

- Image the HDD
- Copy files remotely for analysis

Process Data

- Hash files
- Analyze Signatures

Separate Wheat

- De-NIST or De-NSRL
- Known File Filter (KFF)
- Keyword Searches

Analyze For Relevance

- Good hits or false positives?
- Look at photos, read documents, analyze spreadsheets
- Export files for native analysis
- Bookmark, Flag, or otherwise list useful things

Prepare Report

- Include thumbnails, snapshots, or snippets
- Write-up procedures (Copy/Paste from similar case to speed up workload)
- Attach appendices, lists, etc

Archive Data

- Store images on central NAS
- Shelve HDDs for future use

Classic Anti-Forensic Techniques

- HDD Scrubbing / File Wiping
 - Overwriting areas of disk over and over
- Encryption
 - TrueCrypt, PGP, etc.
- Physical Destruction

These 3 methods are fairly common amongst people like us
In reality, these are used rarely.

Each method implies guilt, and can be dealt with without tech.

Running Tallies on Slides

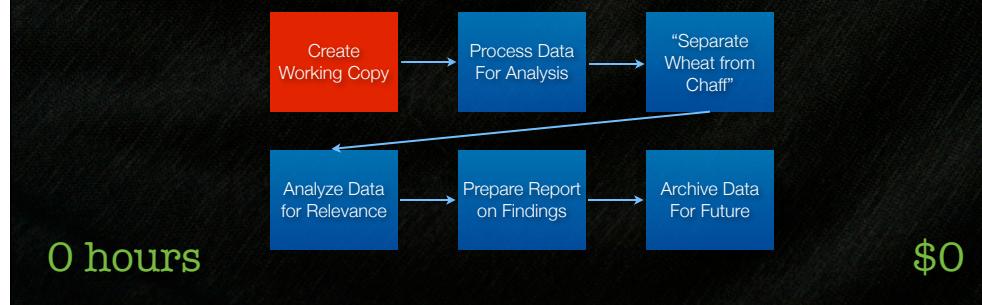
- Tally of # Hours Wasted will be at bottom left
 - Tally of # Dollars Spent will be at bottom right
 - I will assume an average rate of \$300/hr for the digital investigator's time
-
- Red tallies indicate costs for current technique
 - Green tallies show total costs to-date

0 hours ← → \$0

\$300/hr rate is fairly average for junior–intermediate investigators

#1. Create a Working Copy

Confounding the first stage of the process



Copy each device for later analysis
...or copy the file from the remote live machine

AF Technique #1 Data Saturation

- Let's start simple:
 - Own a LOT of media
 - Stop throwing out devices
 - Use each device/container regularly if possible
 - Investigators will need to go through everything



8 hours

\$2,400

Cell Phones

Laptops

Old HDDs

USB Keys

Burned CD/DVDs

Mitigating Data Saturation

- Parallelize the acquisition process
 - More drive duplicators = less total time
 - The limit is your budget.
- Use their hardware against them:
 - Boot from a CD, plug in a USB HDD, mount'n'copy
 - The limit is the # of their machines

8 hours

\$2,400

Incidentally, the # of their machines is typically equal to the number of machines you need to copy!!



9 Machines imaging in parallel to an external USB drive.

Total time = time to image 1 drive.

AF Technique #2 Non-Standard RAID

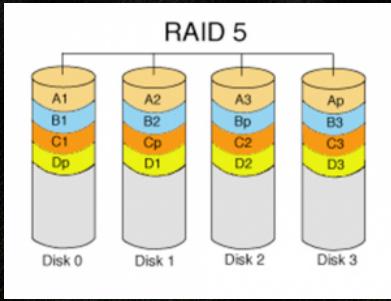
- Common RAIDs share stripe patterns, block sizes, and other parameters
- This hack is simple:
Use uncommon settings
 - Stripe size, stripe order, Endianness
- Use uncommon hardware RAID controllers
(HP Smart Array P420)
- Use firmware with poor Linux support.
Don't flash that BIOS!

8 hours

\$2,400

Non-standard RAID controllers sometimes allow you to choose arbitrary blocksizes
and other parameters that would otherwise be taken care of automatically.

Less damaging for Public sector, can be very expensive for Private sector



- Disk Order (0, 1, 2, 3? 3, 2, 1, 0?)
- Left Synchronous? Right Synchronous?
- Left Asynchronous? Right Asynchronous?
- Big Endian? Little Endian?
- Scott Moulton's DEFCON17 talk about using porn to fix RAID explains this problem well

16 hours

\$4,800

There are so many parameters used by RAID controllers that it can be quite time consuming to try all combinations in order to figure out the exact settings used by the original device

Mitigating Non-Standard RAIDs

- De-RAID volumes on attacker's own system
 - Use boot discs
 - Their hardware reassembles it for you
 - If RAID controller doesn't support Linux, use Windows
Windows-Live CDs work well
 - Image the volume, not the HDDs

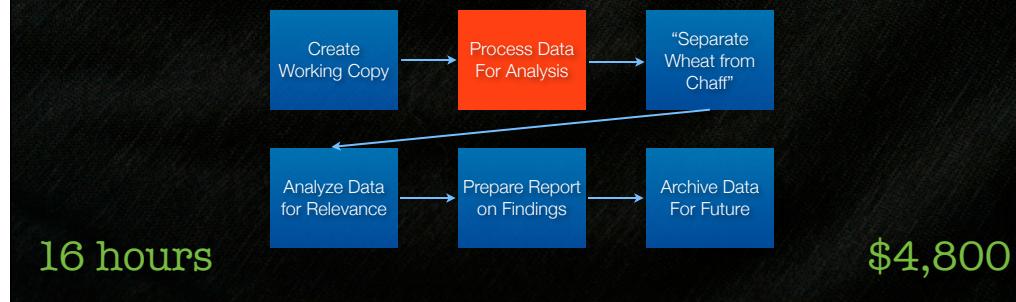
16 hours

\$4,800

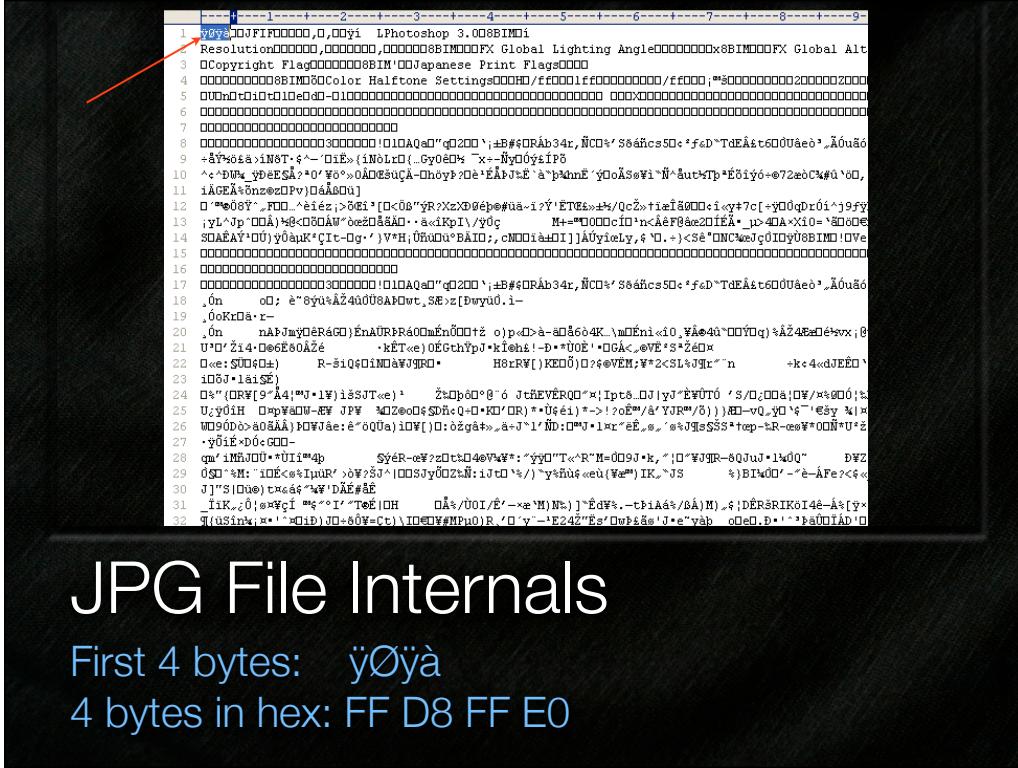
By recombining the RAID array on the attacker's system, their hardware does all the heavy lifting for you.
All you need to worry about is copying the data to your drive.

#2. Process Data for Analysis

Confounding the processing stage



This stage involves:
Hashing
Full-Text Indexing
FileType identification
etc.



JPG File Internals

First 4 bytes: **ÿØÿà**

4 bytes in hex: **FF D8 FF E0**

ZIP Files: PK

EXE Files: MZ

PDF Files: PDF

AF Technique #3 File Signature Masking

- File Signatures are identified by file headers/footers
- “Hollow Out” a file and store your data inside
- Encode data and paste in middle of a binary file
- Transmogrify does this for you

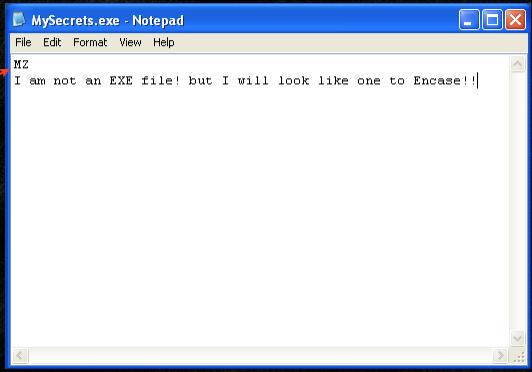
0 hours

\$0

File signatures are identified by the first few bytes
This makes it easy to fake a file match

File Signatures (cont.)

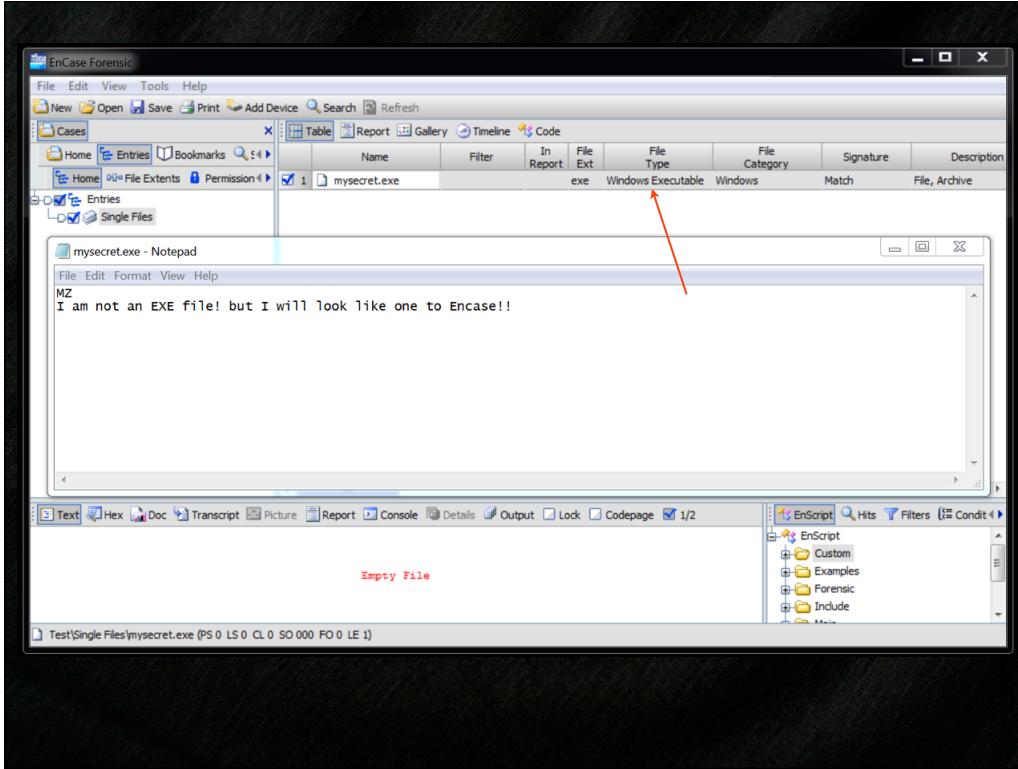
- EXE files begin with bytes MZ
- It's dead easy to make files match their extensions



16 hours

\$4,800

This TXT file shows how easy it is to make a file match its extension despite having contents that are vastly different



Even though this txt file was created in notepad, it is recognized as a
'Windows Executable'
because the first two characters are MZ.

Mitigating File Signature Masking

- Use “Fuzzy Hashing” to identify potentially interesting files
 - Fuzzy Hashing identifies similar but not identical files
 - Chances are, attacker chose a file from his own system to copy/hollow out
 - “Why does this file have a 90% match with notepad.exe?”
- Analyze all “Recent” lists of common apps for curious entries
 - “Why was rundll.dll recently opened in Wordpad?”

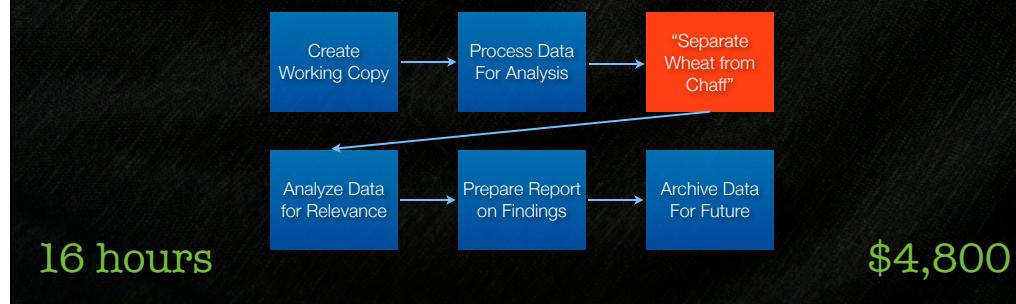
16 hours

\$4,800

Fuzzy Hashing and Recent File analysis can mitigate false file signatures fairly easily by asking simple questions

#3. Separate Wheat from Chaff

Confounding the sifting process



Data Deduplication
Date Filtering
NSRL

Background: NSRL

- National Software Reference Library (NSRL)
- Published by National Institute of Standards and Technology (NIST)
- Huge databases of hash values
- Every dll, exe, hlp, pdf, dat other file installed by every commercial installer
- Used by investigators to filter “typical” stuff
- This process is sometimes called De-NISTing

16 hours

\$4,800

De-NISTing a drive may bring 120GB of data down to about 700MB
Only user-created content will remain

Hundreds of gigabytes can be reduced to a few hundred megabytes

AF Technique #4 NSRL Scrubbing

- Modify all of your system and program files
 - Modify a string or other part of the file
 - For EXEs and DLLs: recalculate and update the embedded CRCs
 - Turn off Data Execution Prevention (DEP) so Windows continues to run
 - NSRL will no longer match ***anything***

12 hours

\$3,600

Most files won't need a lot of work: simply change a character and you're good.

Executable files (DLLs, EXEs) have embedded Cyclical Redundancy Checks (CRCs) that make sure they are still good

You will need to recalculate the CRCs for these files in order to change them in a way that will keep them running

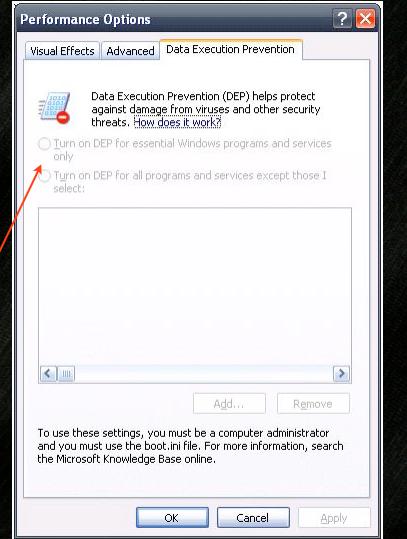
Data Execution Prevention

Validates system files,
Stops unsafe code,
Protects integrity

boot.ini policy_level
/noexecute=AlwaysOff

28 hours

\$8,400



DEP will stop Windows from running if it sees parts of Windows being modified.

So turn it off! You can then run your modified version of Windows without restriction.

Mitigating NSRL Scrubbing

- Search, don't filter
- Identify useful files rather than eliminating useless files
- Use a Whitelist approach instead of a Blacklist

28 hours

\$8,400

Whitelist approach looks for things that match
Blacklist approach suppresses things that don't

Use a whitelist approach

Background: Histograms

- Investigators use histograms to identify which dates have higher-than-average activity
- e.g. VPN Logins, Firewall alerts, even FileCreated times

28 hours

\$8,400

AF Technique #5 Scrambled MACE Times

- All files store multiple timestamps
 - **M**odified - the last write
 - **A**ccessed - the last read
 - **C**reated - the file's birthday 
 - **E**ntry - the last time the MFT entry was updated
- Randomize timestamp of every file
([Timestamp](#) does this)
- Randomize BIOS time regularly via daemon/service
 - Disable LastAccess updates in registry

16 hours

\$4,800

Most of an investigator's "Timeline Assembly" revolve around MACE times
MACE times can be modified easily

A malicious person can modify **EVERY** MACE time across an entire system

LastAccess time can be disabled in two ways:

- In Windows Registry key:
HKEY_LOCAL_MACHINE\SYSTEM
\CurrentControlSet\Control\FileSystem
 - Set DWORD **NtfsDisableLastAccessUpdate** = 1
- Open Command Prompt as Administrator:
FSUTIL behavior set disablelastaccess 1

44 hours

\$13,200

Two ways to suppress “Last Accessed Time” updates

Mitigating Scrambled MAC Times

- Ignore dates on all metadata
- Look for logfiles that write dates as strings
 - Logs are written sequentially
 - BIOS time changes can be identified
- Identify sets of similar times
 - Infer mini timelines for each set
 - Order sets based on what you know of that app

44 hours

\$13,200

Sequential logfiles can help identify timelines

Sequential Log Files: A Timeline

- This log shows 3 sets of similar times
- Order of sets can be identified from this sequential log

```
2026-11-02 11:09:08 Disk Mounted
2026-11-02 11:09:12 Something
2026-11-02 11:09:48 Something Else
2026-11-02 11:10:02 Stuff Happened
2026-11-02 11:10:23 More Stuff Happened
1983-08-23 09:54:45 Log Entry
1983-08-23 09:54:47 Something Occurred
1983-08-23 09:54:58 Another Log Entry
1983-08-23 09:55:03 Stuff Happens
1983-08-23 09:55:09 Stuff!
2003-02-28 15:42:01 More Stuff
2003-02-28 15:42:09 More Stuff
2003-02-28 15:42:12 More Stuff
2003-02-28 15:42:40 More Stuff
2003-02-28 15:42:58 More Stuff
```

44 hours

\$13,200

This logfile shows 3 sets of similar times
It also shows the ordering of each set

The BIOS time was changed twice

Malicious MACE Times

- When all timestamps are scrambled, you know to ignore the values
- If all files appear normal, you will never know if a single file has been updated to appear consistent
- Investigative reports typically cite:
“this time is consistent with that time”
when describing artifacts found during analysis

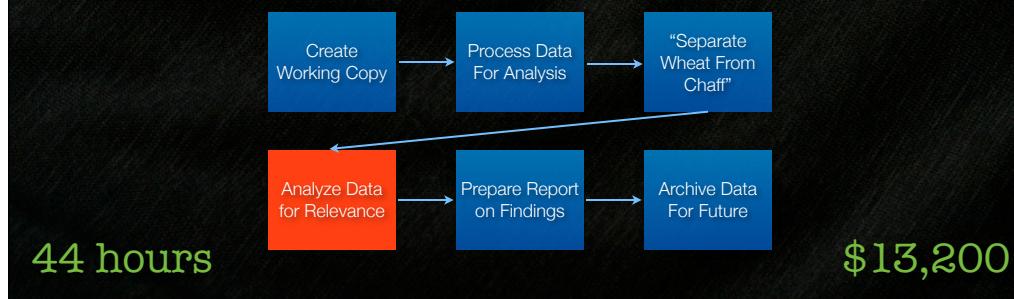
44 hours

\$13,200

Smart investigators never say “This occurred at this time”
They say ‘Logs show it occurred at this time’
and “This time is consistent with these other logs which reference this action”

#4. Analyze Data

Confounding file analysis



When the suite you're using doesn't show you everything you want to see, you typically take the file out of the image to your workstation
You can then use your own app to analyze the file

AF Technique #6 Restricted Filenames

- Even Windows 7 still has holdovers from DOS days:
Restricted filenames
 - CON
 - PRN
 - AUX
 - NUL
 - COM1, COM2, COM3, COM#
 - LPT1, LPT2, LPT#
- Use these filenames liberally

1 hour

\$300

Windows 7 still has parts of DOS in it

This won't take up too much time but will still frustrate the investigator.
He'll likely figure out what's wrong in less than an hour, but will bill a full hour
of work for it.

Creating Restricted Filenames

- Access NTFS volume via UNC
`\host\C$\Folder`
- Call Windows API function `MoveFile` manually from a custom app (`Kernel32.lib`)
- Boot from Linux with NTFS support and `mv` the file

45 hours

\$13,500

You can't just create a file with a restricted name. You need to trick Windows into doing it

Mitigating Restricted Filenames

- Never export files with native filenames
 - Always specify a different name
 - FTK 4 does this by default (1.jpg)
- Export by FileID or other automatically generated name

45 hours

\$13,500

Your analysis machine should go by your rules. You make up the filenames.

AF Technique #7 Circular References

- Folders in folders have typical limit of 255 characters on NTFS
- “Junctions” or “Symbolic Links” can point to a parent
- C:\Parent\Child\Parent\Child....
- Store criminal data in multiple nested files/folders

4 hours

\$1,200

When a circular reference is followed, it could cause programs to enter an infinite loop.

Other programs may detect that the path they're trying to access is > 255 characters and throw an exception

Circular References

- Tools that use HDD images don't bat an eye (FTK4, EnCase)
- Many tools that recursively scan folders are affected by this attack
- “Field Triage” and “Remote Analysis” methodologies are affected

49 hours

\$14,700

Reminder: The 3 Methodologies are:

- * Image Everything, Analyze Later
- * Field Triage to decide what to image
- * Remote Analysis, target only evidence you need

Mitigating Circular References

- Always work from an image
- Be mindful of this attack when dealing with an attacker's live system
- Just knowing about it will help you recognize it

49 hours

\$14,700

AF Technique #8 Broken Log Files

- Many investigators process log files with tools
- These tools use string matching or Regular Expressions
- Use *funny* ASCII characters in custom messages
 - Commas, “quotes” and |pipes| make parsing difficult
 - Use `eLfL` (0x654c664c) in Windows Event Logs

6 hours

\$1,800

`eLfL` is the 4byte header for Windows Event Logs
It marks the start of an eventlog record

Throwing these characters in the middle of a record will confuse some parsers into thinking a new entry has begun

Mitigating Broken Log Files

- Do you need the log? Try to prove your point without it
- Parse the few pertinent records manually and document your methodology
- At worst, write a small app/script to parse it the way you need it to be parsed

51 hours

\$15,300

Zeroing in on the specific records you need is a lot better than parsing the whole log

AF Technique #9 Use Lotus Notes



- NSF files and their .id files are headaches
- There are many tools to deal with NSFs
- Every one of them has its own problems

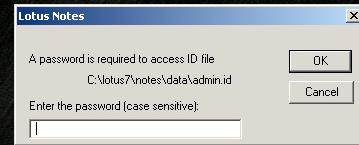
6 hours

\$1,800

Lotus Notes uses NSF files to hold emails, similar to PST files.
ID files include a user ID and an encryption key that can be unlocked
with the user's password
2hrs per custodian

Lotus Notes

- Most apps use IBM's own Lotus Notes dlls/API to work with NSF files
- When opening each encrypted NSF, the API raises the password dialog:
- Examiners/eDiscovery operators must select the user's ID file and type the associated password for every NSF file being processed



57 hours

\$17,100

The password dialog is raised in an interactive context, even when automated

The moment the API is used to open an NSF file, this dialog is presented to the user

This means you can't easily script NSF processing

Mitigating Lotus Notes

- Train yourself on Lotus Notes itself
- Do not rely on NSF conversion tools
- Lotus Notes is the best NSF parser but has its quirks
- Once you know the quirks you can navigate around them

57 hours

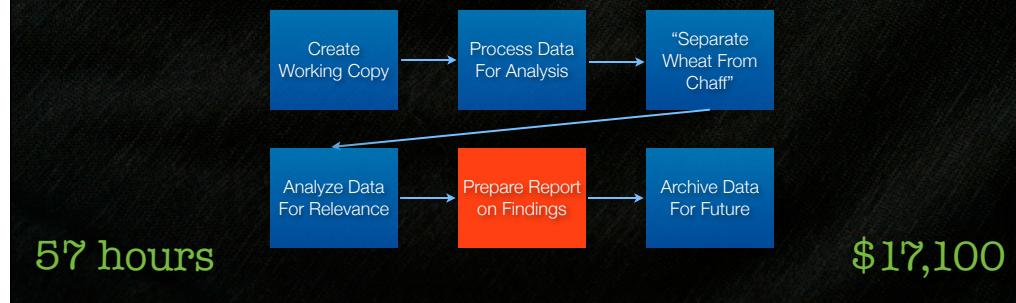
\$17,100

Load up each NSF manually and deal with it using your own keyboard and mouse

Print the notable emails to PDF to be included in your report/affidavit

#5. Report Your Findings

Confounding the reporting process



Reporting didn't seem to have many hacks at first until I started thinking about it....

AF Technique #10 HASH Collisions

- MD5 and SHA1 hashes are used to locate files
- Add dummy data to your criminal files so its MD5 hash matches known good files
- Searching by hash will yield unexpected results
- badfile.doc e4d909c290d0fb1ca068ffaddf22cbd0
goodfile.doc e4d909c290d0fb1ca068ffaddf22cbd0

2 hours

\$600

What if you match your bad stuff with rundll.dll?
NSRL will suppress it!

Hash Collisions

- Of course, this would only be useful in a select few cases:
 - i.e. you stole company data and stored on a volume they could seize/search
- Try explaining why GoodFile.doc and BadFile.doc have identical hashes to judges/justices/arbiters/non-techies
 - could provide just-the-right-amount of 'reasonable doubt'

59 hours

\$17,700

Hash Collisions (cont.)

- Lots of work has been done on this already
- Marc Stevens of the Technische Universiteit Eindhoven developed HashClash for his Masters Thesis in 2008
- Other tools that exploit this are available

59 hours

\$17,700

Most of the research into MD5 collisions is a result of Marc's 2008 paper

Mitigating HASH Collisions

- Use a hash function with fewer collisions (SHA256, Whirlpool)
- Doublecheck your findings by opening each matched file to verify the search was REALLY a success
 - boy would your face be red!

59 hours

\$17,700

Always doublecheck your findings!

Never rely on hash matches to guarantee you've found the file you're looking for

AF Technique #11

Dummy HDD

- PC with an HDD that isn't used
- USB-boot and ignore the HDD for everyday use
- Persist work on cloud/remote machine
- Mimic regular usage of dummy HDD with random writes. Use a daemon/service

3 hours

\$900

Using your computer without a hard drive is very easy nowadays thanks to large removable media

Dummy HDD (cont.)

- Dummy daemon/service can:
 - Retrieve news webpages and write cache to HDD
 - Sync mail with a benign/legit mail account
 - Execute at random intervals
- As long as the HDD has ‘recent’ entries, the investigator will think it’s been used recently

62 hours

\$18,600

Creating a “dummy service” can simulate recent usage of a computer

Mitigating Dummy HDDs

- Always check for USB drives in USB slots AND on motherboards.
They can be SMALL these days...
- Pagefile on USB drive may point to network locations
(if the OS was paging at all...)
- If possible, monitor network traffic before seizure to
detect remote drive location



62 hours

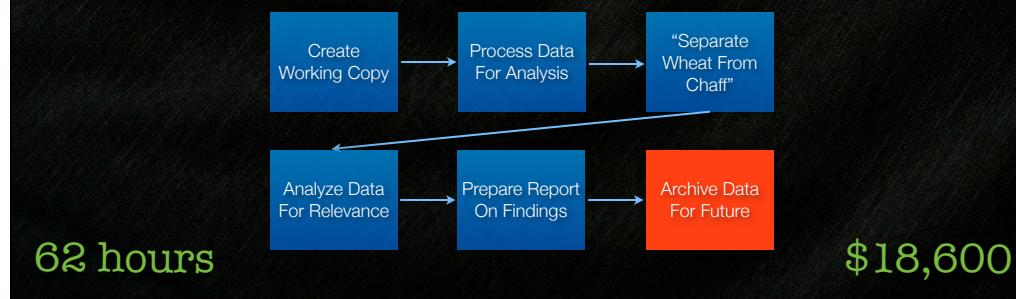
\$18,600

Look on the motherboard itself and identify every USB header.
Follow all USB cables to the external ports on the case.

Don't be fooled!

#6. Archive Data For Future

Confounding the archiving process



In the event a case is challenged in a year or two, firms need to archive data for the future

Technique #1 Data Saturation

- Same as Technique #1
 - The more data you have, the more they need to keep
- We've come full circle



1 hours

\$20/mo per HDD

3 HDDs per month = \$60

3 HDDs per year = \$720

Budget Overrun

- We've taken up roughly 63 hours of an investigator's time
 - That's more than 8 workdays, without overtime
- This extra time was spent trying to image drives, export files, read email, and perform other **menial** tasks
 - The investigator still needs to do his regular work!
- Increased likelihood that opposing council will settle

63 hours

\$18,900 +
\$~720/yr

Questions

- Have you encountered frustration in your examinations?
- How did you deal with it?
- I'd love to hear about it in the speaker Q&A room!

Thanks!

- Thanks DEFCON for letting me speak
- Thanks:
 - Forensic Friends (Josh, Joel, Nick)
 - Family
 - Coworkers
 - You!

Slide Availability

- The slides on your CD are outdated
- You can grab this latest version of these slides from:
[http://www.perklin.ca/~defcon20/
perklin_antiforensics.pdf](http://www.perklin.ca/~defcon20/perklin_antiforensics.pdf)

References

- Berinato, Scott. June 8, 2007. The Rise of Anti-Forensics.
Last accessed on June 12, 2012 from <<http://www.csconline.com/article/221208/the-rise-of-anti-forensics>>
- Max. July 3, 2011. Disk Wiping with dcfldd.
Last accessed on June 12, 2012 from <<http://www.anti-forensics.com/disk-wiping-with-dcfldd>>
- The grugq. Unknown Date. Modern Anti-Forensics.
Last accessed on June 12, 2012 from <<http://sebug.net/paper/Meeting-Documents/syscanhk/Modern%20Anti%20Forensics.pdf>>
- Henry, Paul. November 15, 2007. Anti-Forensics.
Last accessed on June 12, 2012 from <<http://www.techsec.com/pdf/Tuesday/Tuesday%20Keynote%20-%20Anti-Forensics%20-%20Henry.pdf>>
- Garfinkel, Simson. 2007. Anti-Forensics: Techniques, Detection, and Countermeasures.
Last accessed on June 12, 2012 from <<http://simson.net/rei/2007/slides-ICW.pdf>>
- Kessler, Gary. 2007. Anti-Forensics and the Digital Investigator.
Last accessed on June 12, 2012 from <http://www.garykessler.net/library/2007_ADFC_anti-forensics.pdf>
- Hille, S. 2007. Anti-Forensics with a small army of exploits.
Last accessed on June 12, 2012 from <<http://cryptome.org/0003/anti-forensics.pdf>>

References (cont.)

- Dardick, G., La Roche, C., Flanigan, M. 2007. BLOGS: Anti-Forensics and Counter Anti-Forensics.
Last accessed on June 12, 2012 from <<http://igneous.scs.edu.au/proceedings/2007/forensics/21-Dardick%20et.al%20BLOGS%20ANTI-FORENSICS%20and%20COUNTER%20ANTI-FORENSICS.pdf>>
- Hartley, Matthew. August, 2007. Current and Future Threats to Digital Forensics.
Last accessed on June 12, 2012 from <<https://dev.issa.org/Library/Journals/2007/August/Hartley-Current%20and%20Future%20Threats%20to%20Digital%20Forensics.pdf>>
- Perkin, Michael. April 26, 2012. Anti-forensics: Techniques that make our lives difficult, and what we can do to mitigate them.
Presented at HTCIA Ontario Chapter, Brampton, ON, Canada;
- Peron, C., Legary, M.. Digital Anti-Forensics: Emerging trends in data transformation techniques.
Last accessed on June 12, 2012 from <<http://www.seccuris.com/documents/whitepapers/Seccuris-Antiforensics.pdf>>
- Stevens, M. June, 2007. On Collisions for MD5.
Last accessed on June 12, 2012 from <<http://www.win.tue.nl/hashclash/On%20Collisions%20for%20MD5%20-%20M.M.J.%20Stevens.pdf>>
- Foster, J., Liu, V. July 2005. Catch Me If You Can: Exploiting Encase, Microsoft, Computer Associates, and the rest of the bunch...
Last Accessed on June 12, 2012 from <<http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-foster-liu-update.pdf>>
- Moulton, S. July 2009. RAID Recovery: Recover your PORN by sight and sound.
Last Accessed on June 12, 2012 from <http://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-scott_moulton_raid_recovery.pdf>