

How to

Catch an Insider Data Theft

Jonathan Grier

Confidentiality

To preserve client confidentiality, case information (names, places, dates, and settings) has been omitted or altered.

The data and techniques presented have not been altered.



Can you find the data thief?

Data Exfiltration

I've received a number of questions both via e-mail and from customers, asking about data exfiltration. In the vast majority of cases, someone has a system (or an image acquired from a system) and wants to know what data was copied off that system, possibly onto a removable storage device. The fact of the matter is that there are a number of means by which a user can copy data off a system, such as by attaching files to Web-based e-mails, using the built-in File Transfer Protocol (FTP) client, and so forth. When you're looking for indications or "evidence" that files were copied from the system to removable media (e.g., a thumb drive, iPod, etc.), the simple fact is that at this time, there are no apparent artifacts of this process, and you would need to acquire and analyze both pieces of media (i.e., the system that was the source, and the removable media that was the target). Artifacts of a copy operation, such as using the `copy` command or drag-and-drop, are not recorded in the Registry, or within the file system, as far as I and others have been able to determine.

Harlan Carvey, *Windows Forensic Analysis*, 2009

Data Exfiltration

I've received a number of questions both via e-mail and from customers, asking about data exfiltration. In the vast majority of cases, someone has a system (or an image acquired from a system) and wants to know what data was copied off that system, possibly onto a removable storage device. The fact of the matter is that there are a number of means by which a user can copy data off a system, such as by attaching files to Web-based e-mails, using the built-in File Transfer Protocol (FTP) client, and so forth. When you're looking for indications or "evidence" that files were copied from the system to removable media (e.g., a thumb drive, iPod, etc.), the simple fact is that at this time, there are no apparent artifacts of this process, and you would need to acquire and analyze both pieces of media (i.e., the system that was the source, and the removable media that was the target). Artifacts of a copy operation, such as using the *copy* command or drag-and-drop, are not recorded in the Registry, or within the file system, as far as I and others have been able to determine.

Harlan Carvey, *Windows Forensic Analysis*, 2009

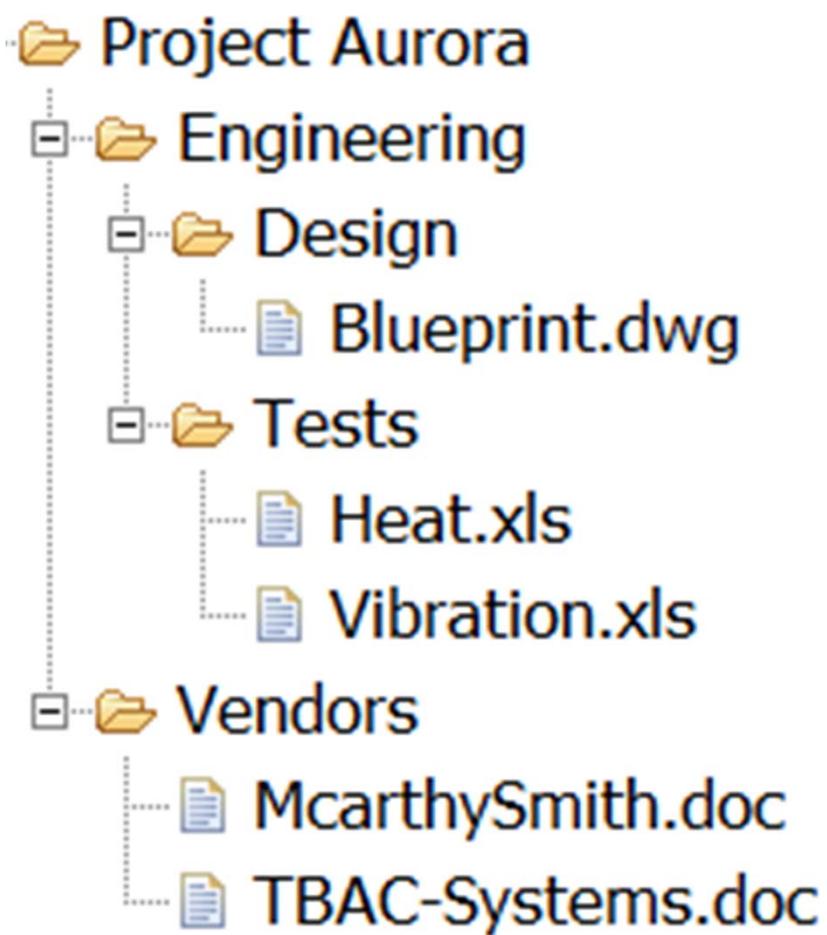
No Artifacts = No Forensics

Data Exfiltration

I've received a number of questions both via e-mail and from customers, asking about data exfiltration. In the vast majority of cases, someone has a system (or an image acquired from a system) and wants to know what data was copied off that system, possibly onto a removable storage device. The fact of the matter is that there are a number of means by which a user can copy data off a system, such as by attaching files to Web-based e-mails, using the built-in File Transfer Protocol (FTP) client, and so forth. When you're looking for indications or "evidence" that files were copied from the system to removable media (e.g., a thumb drive, iPod, etc.), the simple fact is that at this time, there are no apparent artifacts of this process, and you would need to acquire and analyze both pieces of media (i.e., the system that was the source, and the removable media that was the target). Artifacts of a copy operation, such as using the `copy` command or drag-and-drop, are not recorded in the Registry, or within the file system, as far as I and others have been able to determine.

Harlan Carvey, *Windows Forensic Analysis*, 2009

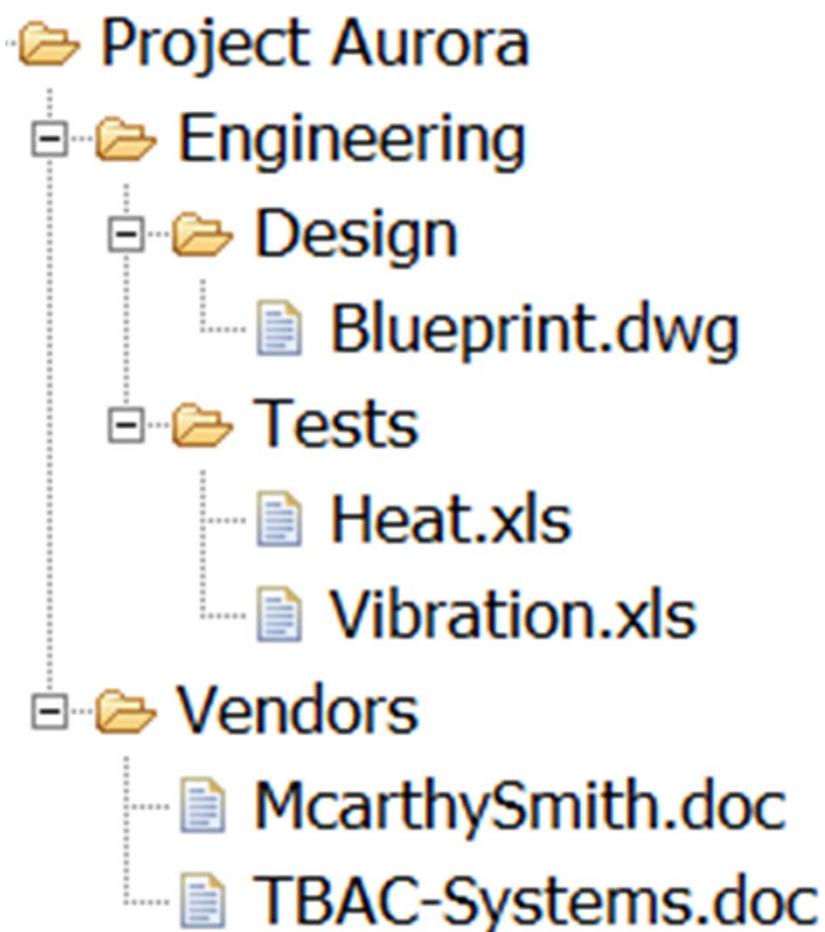
No Artifacts = No Forensics ???



Access timestamps updates during:

Routine access

Project Aurora	1.	9:13:01 AM
Engineering	2.	9:13:03 AM
Design		
Blueprint.dwg	6.	9:21:47 AM
Tests	3.	9:13:04 AM
Heat.xls		
Vibration.xls	4.	9:13:06 AM
Vendors		
McarthySmith.doc	5.	9:17:25 AM
TBAC-Systems.doc		



Access timestamps updates during:

Copying a folder

1.	9:13:01 AM	Project Aurora
2.	9:13:01 AM	Engineering
3.	9:13:01 AM	Design
4.	9:13:01 AM	Blueprint.dwg
5.	9:13:03 AM	Tests
6.	9:13:03 AM	Heat.xls
7.	9:13:04 AM	Vibration.xls
8.	9:13:05 AM	Vendors
9.	9:13:05 AM	McarthySmith.doc
10.	9:13:05 AM	TBAC-Systems.doc

Routine access

1.	9:13:01 AM	Project Aurora
2.	9:13:03 AM	Engineering
3.	9:13:04 AM	Tests
4.	9:13:06 AM	Vibration.xls
5.	9:17:25 AM	McarthySmith.doc
6.	9:21:47 AM	Blueprint.dwg

Copying Folders

Routine Access

Nonselective

All subfolders and files accessed

Selective

Temporally continuous

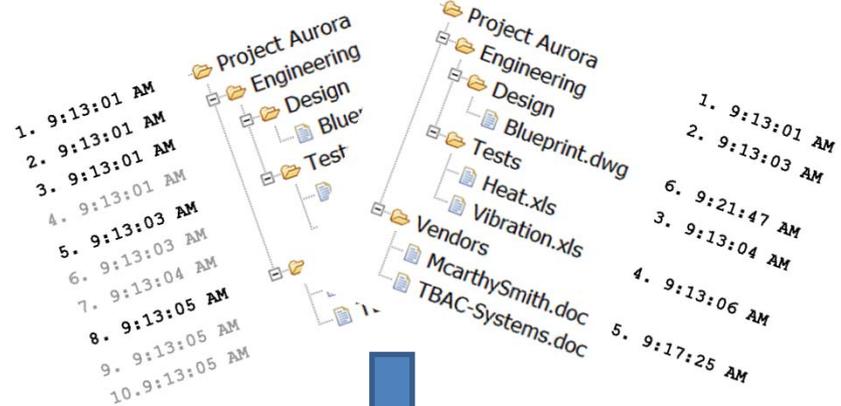
Temporally irregular

Recursive

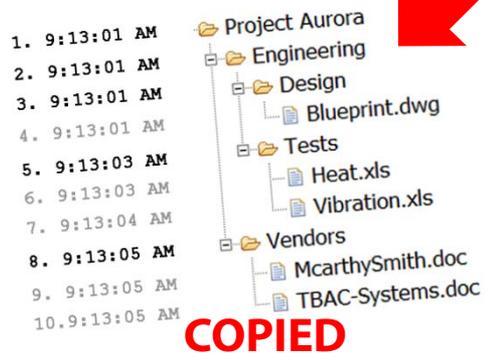
Random order

Directory accessed
before its files

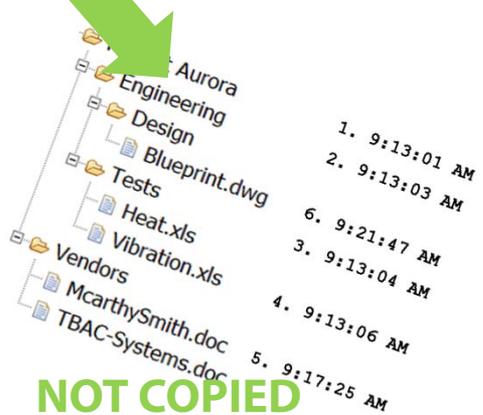
Files can be accessed
without directory



Copying Folders	Routine Access
Nonselective <i>All subfolders and files accessed</i>	Selective
Temporally continuous	Temporally irregular
Recursive	Random order
Directory accessed before its files	Files can be accessed without directory



COPIED



NOT COPIED

No Artifacts Yes Forensics

Copying Folders	Routine Access
Nonselective All subfolders and files accessed	Selective
Temporally continuous	Temporally irregular
Recursive	Random order
Directory accessed before its files	Files can be accessed without directory

“slap-your-head-and-say-'doh-wish-I'd-thought-of-that”

-- an anonymous reviewer

Not so fast...

1. Timestamps are overwritten *very quickly*
2. There are other nonselective, recursive activities (besides copying)

Not so fast...

1. Timestamps are overwritten *very quickly*

Can we use this method months later?

On a heavily used system?

Won't most of the timestamps have been overwritten?

Not so fast...

1. Timestamps are overwritten *very quickly*

YES! Can we use this method months later?

YES! On a heavily used system?

Not really! Won't most of the timestamps have been overwritten?

Two observations:

1. Timestamps values can *increase*, but never *decrease*.
2. A lot of files just collect dust.
Most activity is on a minority of files.

The vast majority of files on two fairly typical Web servers have not been used at all in the last year. Even on an extraordinarily heavily used (and

Table 1.1 *Percentage of files read or executed recently for a number of Internet servers*

	www.things.org	www.fish.com	news.earthlink.net
Over one year:	76.6	75.9	10.9
Six months to one year:	7.6	18.6	7.2

Farmer & Venema, *Forensic Discovery*, 2005

Pareto Principle

- 80% of your profits come from 20% of your customers
 - 80% of your complaints come from 20% of the customers
 - 80% of your profits come from 20% of your products
 - 80% of your sales come from 20% of your salespeople
- http://en.wikipedia.org/wiki/Pareto_principle



Mathematics Vol. 1, No. 2: 226-251

A Brief History of
Generative Models for
Power Law and Lognormal
Distributions

Mitzenmacher

At t_{copying} :

- All files have `access_timestamp = tcopying`

At t_{copying} :

- All files have $\text{access_timestamp} = t_{\text{copying}}$

Several weeks later:

- All files have $\text{access_timestamp} \geq t_{\text{copying}}$

At t_{copying} :

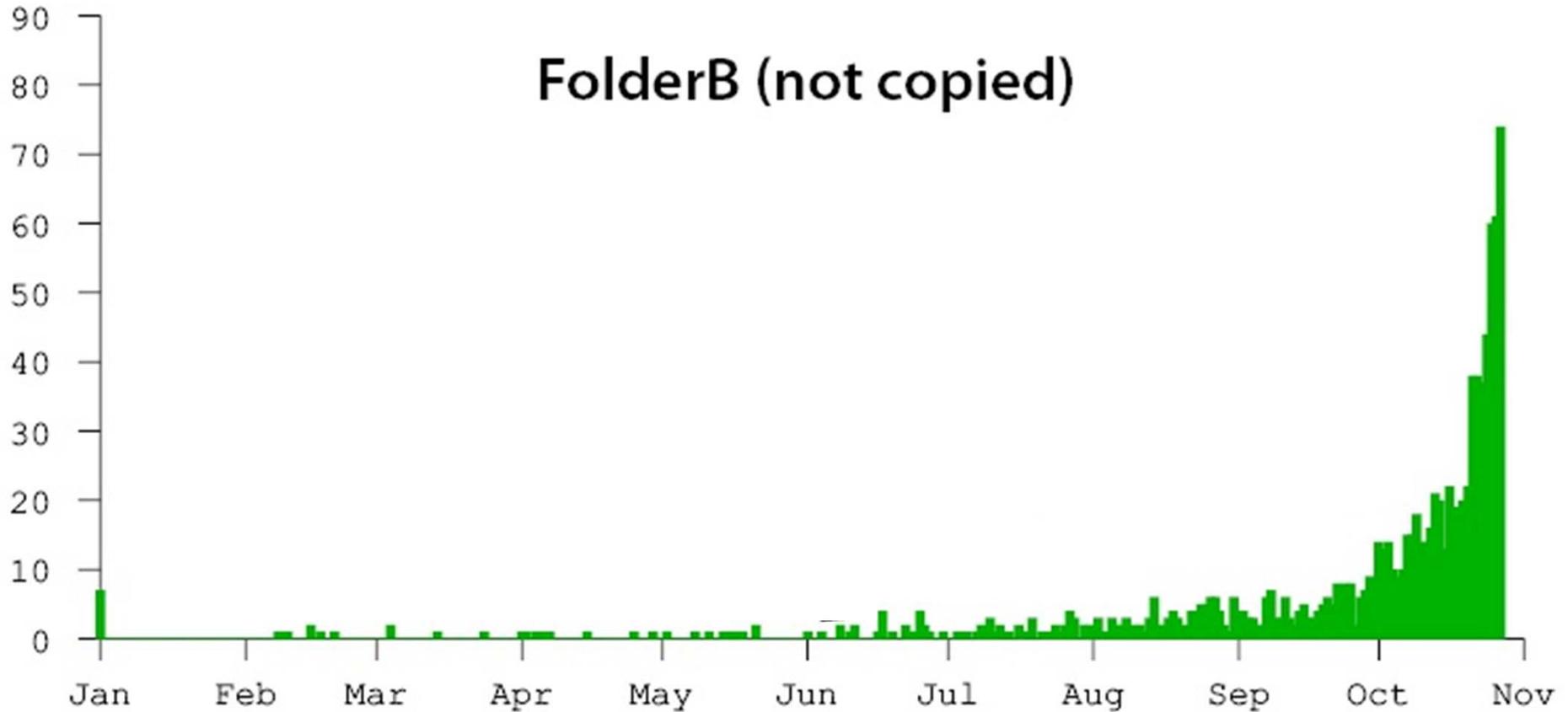
- All files have $\text{access_timestamp} = t_{\text{copying}}$

Several weeks later:

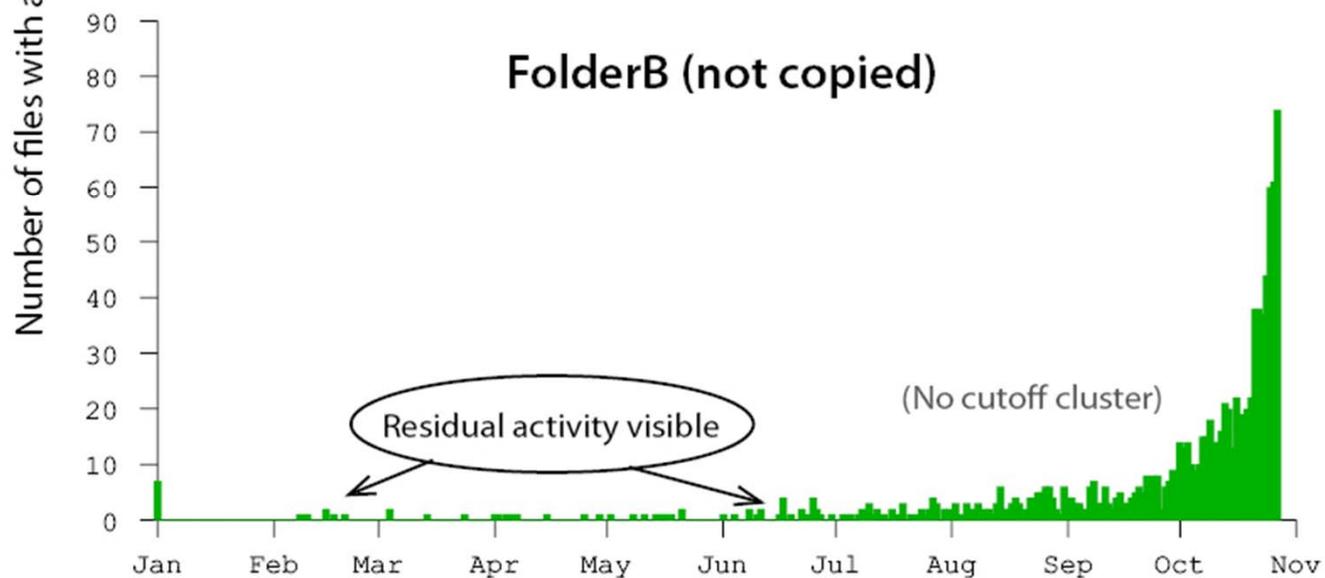
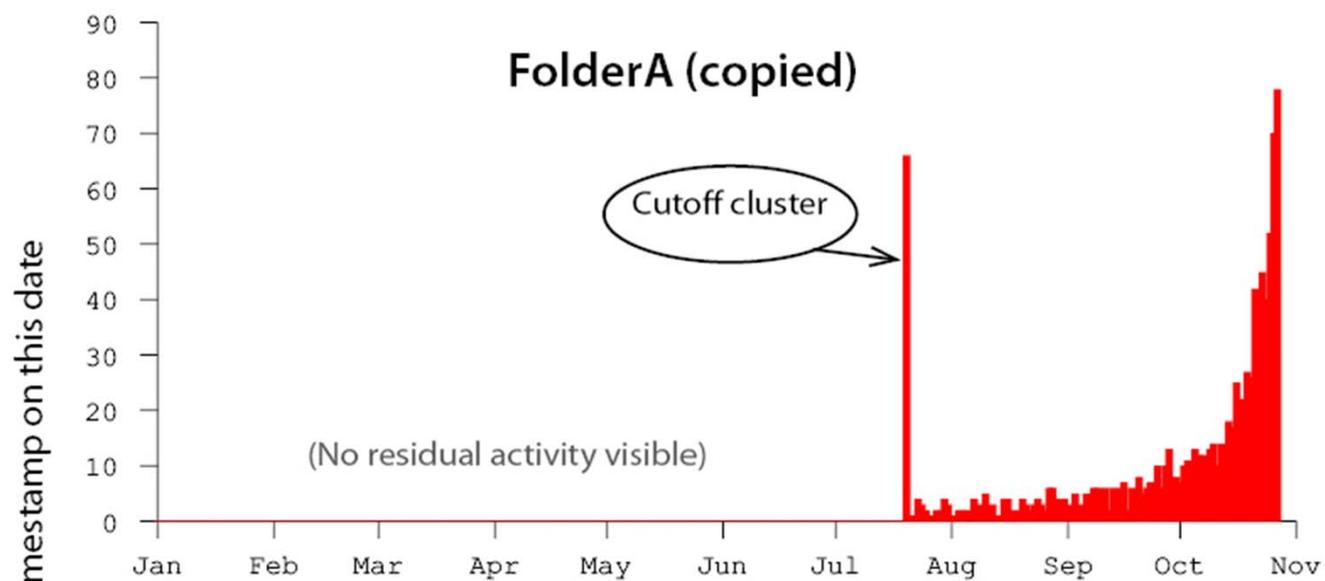
- All files have $\text{access_timestamp} \geq t_{\text{copying}}$
- **Many** files still have $\text{access_timestamp} = t_{\text{copying}}$

Histogram of access timestamps

FolderB (not copied)



After 300 days of simulated activity



Copying creates a

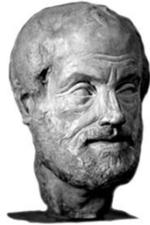
cutoff cluster

cutoff – No file has timestamp $< t_{\text{cluster}}$

cluster – Many files have timestamp $= t_{\text{cluster}}$

Aren't there other recursive access patterns besides copying?

Affirming the
consequent



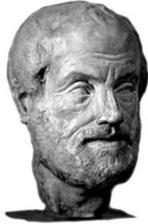
$A \rightarrow B$ doesn't prove $B \rightarrow A$.

The *absence* of a cutoff cluster can disprove copying, but the *existence* can't prove copying.

Perhaps they ran `grep`.

Indeed, there are!

Affirming the consequent



$A \rightarrow B$ doesn't prove $B \rightarrow A$.

VS.



Abductive reasoning

An unusual observation supports inferring a likely cause.

The *absence* of a cutoff cluster can disprove copying, but the *existence* can't prove copying.

Perhaps they ran `grep`.

Who's trying to *prove* anything?
Investigate! One clue leads to another until the case unravels.

Indeed!
Check if `grep` is installed, if they've ever run it before, or after, on any folder.
Check why they were still in the building at 11 PM.

Forensics

WHY ~~PROGRAMMING~~ IS A GOOD MEDIUM FOR ~~EXPRESSING~~
POORLY UNDERSTOOD AND SLOPPILY-FORMULATED IDEAS.

-- Marvin Minsky, MIT, 1967

Forensics

WHY ~~PROGRAMMING~~ IS A GOOD MEDIUM FOR ~~EXPRESSING~~ ^{investigating}
POORLY UNDERSTOOD AND SLOPPILY-FORMULATED IDEAS.

-- Marvin Minsky, MIT, 1967

Our general philosophy recommends greater understanding instead of higher levels of certainty, which could potentially make such methodology more suspect in a court of law. Paradoxically, however, the uncertainty—primarily in the data collection methods—can actually give a greater breadth of knowledge and more confidence in any conclusions

Farmer & Venema, *Forensic Discovery*, 2005

An actual investigation...

Part II:

**Now for the real
world...**

NOISE

OpenSolaris `cp` command source code

```
    if (m < 0) {
        (void) fprintf(stderr, gettext(
            "cp: cannot change owner and "
            "group of %s:"), target);
        perror("");
    }
} else {
    /*
     * Copy the file.  If it happens to be a
     * symlink, copy the file referenced
     * by the symlink.
     */
    fi = open(source, O_RDONLY);
    if (fi < 0) {
        (void) fprintf(stderr,
            gettext("%s: cannot open %s: "),
            and_source);
```

Notice anything?

```
    if (m < 0) {
        (void) fprintf(stderr, gettext(
            "cp: cannot change owner and "
            "group of %s:"), target);
        perror("");
    }
} else {
    /*
     * Copy the file.  If it happens to be a
     * symlink, copy the file referenced
     * by the symlink.
     */
    fi = open(source, O_RDONLY);
    if (fi < 0) {
        (void) fprintf(stderr,
            gettext("%s: cannot open %s: "),
            and_source);
```

Notice anything?

```
    if (m < 0) {
        (void) fprintf(stderr, gettext(
            "cp: cannot change owner and "
            "group of %s:"), target);
        perror("");
    }
} else {
    /*
     * Copy the file.  If it happens to be a
     * symlink, copy the file referenced
     * by the symlink.
     */
    fi = open(source, O_RDONLY);
    if (fi < 0) {
        (void) fprintf(stderr,
            gettext("%s: cannot open %s: "),
            and source);
```

OpenSolaris `cp` command source code

`writelfile()` function

```
    /*
    * Mmap time!
    */
    if ((cp = mmap((caddr_t)NULL, mapsize, PROT_READ,
        MAP_SHARED, fi, (off_t)0)) == MAP_FAILED)
        mapsize = 0;    /* can't mmap today */
} else
    mapsize = 0;

if (mapsize != 0) {
    offset = 0;

    for (;;) {
        nbytes = write(fo, cp, mapsize);
        /*
        * if we write less than the mmaped size it's due to a
        * media error on the input file or out of space on
        * the output file.  So, try again, and look for errno.
        */
    }
}
```

CopyFile function

msdn

Copies an existing file to a new file.

The **CopyFileEx** function provides two additional capabilities.

CopyFileEx can call a specified callback function each time a portion of the copy operation is completed, and **CopyFileEx** can be canceled during the copy operation.

To perform this operation as a transacted operation, use the **CopyFileTransacted** function.

Syntax

C++

```
BOOL WINAPI CopyFile(  
    _In_ LPCTSTR lpExistingFileName,  
    _In_ LPCTSTR lpNewFileName,  
    _In_ BOOL bFailIfExists  
);
```

Is all lost
(on Windows at least)

?

Tree

- Windows
 - addins
 - AppCompat
 - AppPatch
 - assembly

Properties

File List

Name	
addin	
AppC	
AppP	
asser	
Boot	
Branc	
CSC	
Cursc	
00	30 00 0
10	10 00 0
20	00 00 0
30	05 00 0

General

Name	Windows
File Class	Directory
File Size	56
Physical Size	56
Date Accessed	7/10/2012 6:00:33 PM
Date Created	7/14/2009 3:20:08 AM
Date Modified	7/10/2012 6:00:33 PM
Encrypted	False
Compressed	False
Actual File	True
Alternate Data Stream	2

DOS Attributes

NTFS Information

MFT Record Number	17,129 (1754009664d)
Record date	7/10/2012 6:00:33 PM
Resident	True
Offline	False
Sparse	False
Temporary	False
Owner SID	S-1-5-80-956008885-3
Group SID	S-1-5-80-956008885-3

NTFS Access Control Entry

ACE Type	Allow Access
Inheritable	False
SID	S-1-5-80-956008885-3
Access Mask	001f01ff

**a Directory
is also
a File!**

The screenshot shows a Windows Explorer window with the 'Tree' pane on the left and the 'Properties' pane on the right. The 'Tree' pane shows a folder structure under 'Windows' with subfolders: 'addins', 'AppCompat', 'AppPatch', and 'assembly'. The 'Properties' pane is open for the 'Windows' folder, showing the following details:

General	
Name	Windows
File Class	Directory
File Size	56
Physical Size	56
Date Accessed	7/10/2012 6:00:33 PM
Date Created	7/14/2009 3:20:08 AM
Date Modified	7/10/2012 6:00:33 PM
Encrypted	False
Compressed	False
Actual File	True
Alternate Data Stream	2

DOS Attributes	
Attributes	NTFS

NTFS Information	
MFT Record Number	17,129 (1754009664d)
Record date	7/10/2012 6:00:33 PM
Resident	True
Offline	False
Sparse	False
Temporary	False
Owner SID	S-1-5-80-956008885-3
Group SID	S-1-5-80-956008885-3

NTFS Access Control Entry	
ACE Type	Allow Access
Inheritable	False
SID	S-1-5-80-956008885-3
Access Mask	001f01ff

Filter...



ACCURACY?

Who needs
ACCURACY?

Part III:

Applying Stochastic Forensics

Filter

&

Plot

Filter

1. By folder

Filter

1. By folder

2. Directories versus Files

Filter

1. By folder
2. Directories versus Files
- 3. Permissions**

Filter

1. By folder
2. Directories versus Files
3. Permissions
- 4. Other**

Plot

Our visual cognition is
amazingly robust

Ploticus: <http://ploticus.sourceforge.net>

Interpret & Advance

No Cluster?

**Strong evidence
of *no* copying**

Found Cluster?

- 1. Check control folders**
- 2. Search for causes**
- 3. Fingerprint it**

Found Cluster?

**A cluster defines a tight
*window of opportunity.***

**Use it to *propel the
investigation forward.***

Part IV:

**New Paradigms of
Forensics**

Classical Forensics:

Look at the
Surviving Data



Reconstruct
Previous Data



This previous data
is our deliverable.

Classical Forensics:

Look at the Surviving Data → Reconstruct Previous Data → This previous data is our deliverable.

Stochastic Forensics:

What do I want to know about? → What behavior is associated? → How does that behavior affect the system? → Measure those effects. Draw a (quantifiable) inference.

Digital Forensics Research: The Next 10 Years

Simson L. Garfinkel
Naval Postgraduate School
May 10, 2010

Digital Forensics Research: The Good, the Bad, and the Unaddressed

by Nicole L. Beebe, Ph.D.
5th Annual IFIP WG 11.9
January 27, 2009

Leading researchers have called to move from:
“What data can we find?”
To:
“What did this person do?”

Forensics

WHY ~~PROGRAMMING~~ IS A GOOD MEDIUM FOR ~~EXPRESSING~~
POORLY UNDERSTOOD AND SLOPPILY-FORMULATED IDEAS.

-- Marvin Minsky, MIT, 1967

Forensics

WHY ~~PROGRAMMING~~ IS A GOOD MEDIUM FOR ~~EXPRESSING~~ ^{investigating} POORLY UNDERSTOOD AND SLOPPILY-FORMULATED IDEAS.

-- Marvin Minsky, MIT, 1967

Our general philosophy recommends greater understanding instead of higher levels of certainty, which could potentially make such methodology more suspect in a court of law. Paradoxically, however, the uncertainty—primarily in the data collection methods—can actually give a greater breadth of knowledge and more confidence in any conclusions

Farmer & Venema, *Forensic Discovery*, 2005

**Questions?
Comments?
Want More Info?**

**Please speak to me,
here at CEIC
or jgrier at grierforensics com.**