I’m your MAC(b) Daddy

Grayson Lenik – Trustwave
@handlefree

Bio

- Security Consultant for Trustwave's Spiderlabs.
- Author of the Digital Forensics blog "An Eye on Forensics"
- GIAC Certified Forensic Analyst (GCFA)
- Microsoft Certified Systems Engineer (MCSE)
- Qualified Security Assessor (QSA)

Agenda

- What are MACB times?
- Where are they stored?
- What is a forensic timeline?
- Why is it useful?
- Why do it may way?
- Doing it my way
- My way just got easier
- Timestamp alteration (Timestomping)
- Defeating timestamp alteration
- Case studies and examples.
- Tools
- Conclusion

MAC(b) Times

What do they stand for?

The MAC(b) times are derived from file system metadata and they stand for:

Modified

Accessed

Changed ($MFT Modified)

and

Birth (file creation time)

The (b) is in parentheses because not all file systems record a birth time. For the purposes of this presentation I focus on the NTFS file system as it is the most common file system we see when working investigations.

Where are they stored?

Two places, both are located in the Master File Table ($MFT):

The first is $STANDARD_INFO Attribute.

$STANDARD_INFO ($SI) stores file metadata such as flags, the file SID, the file owner and a set of MAC(b) timestamps.

$STANDARD_INFO is the timestamp collected by Windows explorer, fls, mactime, timestomp, find and the other utilities related to the display of timestamps.

*http://blogs.technet.com/b/ganand/archive/2008/02/19/ntfs-time-stamps-file-created-in-1601-modified-in-1801-and-accessed-in-2008.aspx

Where are they stored?

The second attribute is $FILE_NAME Att.

- $FILE_NAME ($FN) contains the filename in Unicode and another set of MAC(b) timestamps.

The difference?

$STANDARD_INFO can be modified by user level processes like timestomp.

$FILE_NAME can only be modified by the **system kernel**. There are no known anti-forensics utilities that can accomplish this.

What is a forensic timeline?

- A forensic timeline is a string of digital events sorted into a format that is easily readable by a human being.
- It can contain events from a single source, like the file system itself, or events from multiple sources like the system registry, log files, event logs, etc…
- The only way I know of to get a 30,000 foot view of the events that were taking place on a machine during a given time frame.

Why is it useful?

Forensic timelines are infinitely useful at pinpointing all (or most) intruder activity at a given point in time. It is also an excellent place to generate initial case leads and keywords.

- Periods of intruder activity stick out like a sore thumb.
- Identifying a starting point of intrusion is invaluable to finding other pieces of malware and more of the total activity that took place on the targeted system.
- When you add in registry timeline info you get a more complete picture of code being dropped and services being created. In some cases definitive proof of which user id was used to compromise the system.

Why do it my way?

It's fast…

It's easy…

It's easily searchable so you can use your scalpel instead of your shotgun.

It's free! (more or less)

- I can generate a timeline including full file system data and all the registry hives, search it for keywords and identify time altered files before any popular GUI based forensic utilities have loaded and verified an image.
- And in a few minutes, I'm going to show you how!

Doing it my way.

Generate a bodyfile with “fls”:

Fls is open source and is available as part of The Sleuthkit (TSK)*

- *Fls –m C: -f ntfs –r \\.\Z: >fs_bodyfile*
  - m Output in standard format (timemachine format)
  - C: name the mountpoint (could be D:, /var, etc..)
  - -f <filesystem type>
  - -r display directories recursively
  - \\.\Z: point it at a logical device
  - > dump it out to a file called fs_bodyfile

  - This can be done on a live file system by using F-response (the only part of this that’s not free, but **Oh so worth it.**)

*** http://www.sleuthkit.org/sleuthkit/download.php**

Doing it my way.

Fls can also be run against a static (postmortem) image:

Like so:

- *Fls –m C: –o 63 –r <path_to_image> >fs_bodyfile*
  - -o <sector offset where the file system starts in the image.>

Or against a split image:

Like so:

- *Fls –m C: -o 63 –r <path_to_image>, <path to image.002>, <path to image.003> >fs_bodyfile*
  - -o <sector offset where the file system starts in the image.>

Doing it my way.

Here's a snippet of what a bodyfile looks like:

It's not very user friendly.   We're going to see that inetmgr.exe again…

```
0|C:/WINDOWS/system32/imagehlp.dll|45808-128-
    1|r/rrwxrwxrwx|0|0|150016|1294398049|1202272697|1288808002|1181611484
0|C:/WINDOWS/system32/imgutil.dll|53632-128-
    1|r/rrwxrwxrwx|0|0|34816|1294399529|1236501098|1288808002|1236501098
0|C:/WINDOWS/system32/inetcomm.dll|39531-128-
    1|r/rrwxrwxrwx|0|0|695808|1295017350|1276070390|1288808002|1181612350
0|C:/WINDOWS/system32/inetcpl.cpl|53636-128-
    1|r/rrwxrwxrwx|0|0|1469440|1294399448|1284055250|1294399566|123650129
    20|C:/WINDOWS/system32/inetinfo.chm|46996-128-
    0|r/rrwxrwxrwx|0|0|929955|1099455632|1099455632|1293869074|1099455632
0|C:/WINDOWS/system32/inetmgr.exe|47066-128-
    3|r/rrwxrwxrwx|0|0|1301406|1295017321|1068995339|1294399979|106899533
    9
```

Doing it my way

Turning fls output into something useful and human-readable:

Mactime.pl is also available as part of TSK.

Perl mactime.pl –d –b fs_bodyfile >fs_timeline.csv

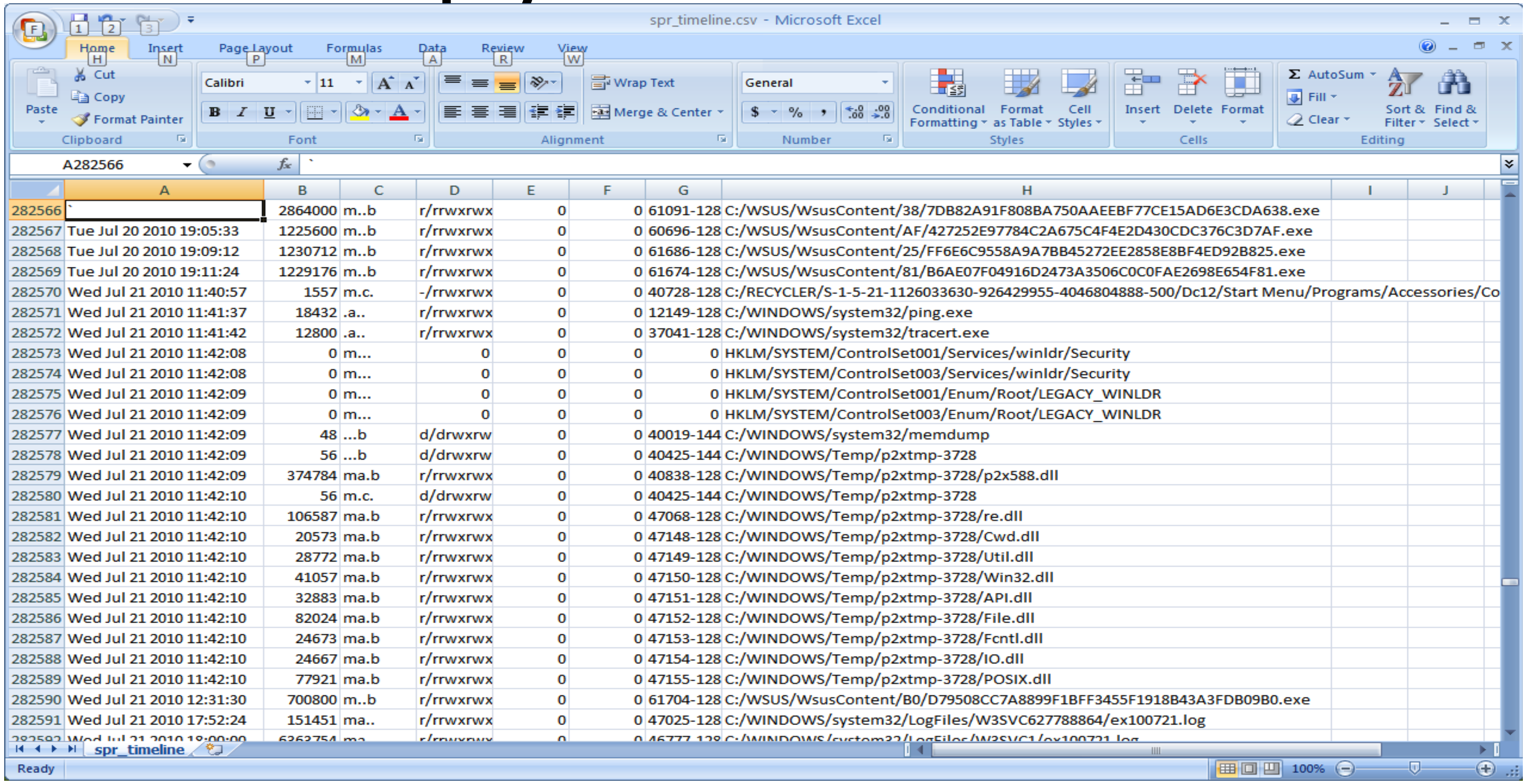
-d output in comma delimited format

-b <path to bodyfile>

> output to a file called fs_timeline.csv

Doing it my way.

Your timeline displayed in Excel.


spr_timeline.csv - Microsoft Excel
Home Insert Page Layout Formulas Data Review View
Paste Cut Copy Format Painter Clipboard
Calibri 11 Font
Wrap Text Merge & Center Alignment
General Number
Conditional Formatting Format as Table Cell Styles Styles
Insert Delete Format Cells
AutoSum Fill Clear Sort & Filter Find & Select Editing
A282566
spr_timeline
Ready 100%

	A	B	C	D	E	F	G	H	I	J
282566	`	2864000	m..b	r/rrwxrwx	0	0	61091-128	C:/WSUS/WsusContent/38/7DB82A91F808BA750AAEEBF77CE15AD6E3CDA638.exe		
282567	Tue Jul 20 2010 19:05:33	1225600	m..b	r/rrwxrwx	0	0	60696-128	C:/WSUS/WsusContent/AF/427252E97784C2A675C4F4E2D430CDC376C3D7AF.exe		
282568	Tue Jul 20 2010 19:09:12	1230712	m..b	r/rrwxrwx	0	0	61686-128	C:/WSUS/WsusContent/25/FF6E6C9558A9A7BB45272EE2858E8BF4ED92B825.exe		
282569	Tue Jul 20 2010 19:11:24	1229176	m..b	r/rrwxrwx	0	0	61674-128	C:/WSUS/WsusContent/81/B6AE07F04916D2473A3506C0C0FAE2698E654F81.exe		
282570	Wed Jul 21 2010 11:40:57	1557	m.c.	-/rrwxrwx	0	0	40728-128	C:/RECYCLER/S-1-5-21-1126033630-926429955-4046804888-500/Dc12/Start Menu/Programs/Accessories/Co		
282571	Wed Jul 21 2010 11:41:37	18432	.a..	r/rrwxrwx	0	0	12149-128	C:/WINDOWS/system32/ping.exe		
282572	Wed Jul 21 2010 11:41:42	12800	.a..	r/rrwxrwx	0	0	37041-128	C:/WINDOWS/system32/tracert.exe		
282573	Wed Jul 21 2010 11:42:08	0	m...	0	0	0	0	HKLM/SYSTEM/ControlSet001/Services/winldr/Security		
282574	Wed Jul 21 2010 11:42:08	0	m...	0	0	0	0	HKLM/SYSTEM/ControlSet003/Services/winldr/Security		
282575	Wed Jul 21 2010 11:42:09	0	m...	0	0	0	0	HKLM/SYSTEM/ControlSet001/Enum/Root/LEGACY_WINLDR		
282576	Wed Jul 21 2010 11:42:09	0	m...	0	0	0	0	HKLM/SYSTEM/ControlSet003/Enum/Root/LEGACY_WINLDR		
282577	Wed Jul 21 2010 11:42:09	48	...b	d/drwxrw	0	0	40019-144	C:/WINDOWS/system32/memdump		
282578	Wed Jul 21 2010 11:42:09	56	...b	d/drwxrw	0	0	40425-144	C:/WINDOWS/Temp/p2xtmp-3728		
282579	Wed Jul 21 2010 11:42:09	374784	ma.b	r/rrwxrwx	0	0	40838-128	C:/WINDOWS/Temp/p2xtmp-3728/p2x588.dll		
282580	Wed Jul 21 2010 11:42:10	56	m.c.	d/drwxrw	0	0	40425-144	C:/WINDOWS/Temp/p2xtmp-3728		
282581	Wed Jul 21 2010 11:42:10	106587	ma.b	r/rrwxrwx	0	0	47068-128	C:/WINDOWS/Temp/p2xtmp-3728/re.dll		
282582	Wed Jul 21 2010 11:42:10	20573	ma.b	r/rrwxrwx	0	0	47148-128	C:/WINDOWS/Temp/p2xtmp-3728/Cwd.dll		
282583	Wed Jul 21 2010 11:42:10	28772	ma.b	r/rrwxrwx	0	0	47149-128	C:/WINDOWS/Temp/p2xtmp-3728/Util.dll		
282584	Wed Jul 21 2010 11:42:10	41057	ma.b	r/rrwxrwx	0	0	47150-128	C:/WINDOWS/Temp/p2xtmp-3728/Win32.dll		
282585	Wed Jul 21 2010 11:42:10	32883	ma.b	r/rrwxrwx	0	0	47151-128	C:/WINDOWS/Temp/p2xtmp-3728/API.dll		
282586	Wed Jul 21 2010 11:42:10	82024	ma.b	r/rrwxrwx	0	0	47152-128	C:/WINDOWS/Temp/p2xtmp-3728/File.dll		
282587	Wed Jul 21 2010 11:42:10	24673	ma.b	r/rrwxrwx	0	0	47153-128	C:/WINDOWS/Temp/p2xtmp-3728/Fcntl.dll		
282588	Wed Jul 21 2010 11:42:10	24667	ma.b	r/rrwxrwx	0	0	47154-128	C:/WINDOWS/Temp/p2xtmp-3728/IO.dll		
282589	Wed Jul 21 2010 11:42:10	77921	ma.b	r/rrwxrwx	0	0	47155-128	C:/WINDOWS/Temp/p2xtmp-3728/POSIX.dll		
282590	Wed Jul 21 2010 12:31:30	700800	m..b	r/rrwxrwx	0	0	61704-128	C:/WSUS/WsusContent/B0/D79508CC7A8899F1BFF3455F1918B43A3FDB09B0.exe		
282591	Wed Jul 21 2010 17:52:24	151451	ma..	r/rrwxrwx	0	0	47025-128	C:/WINDOWS/system32/LogFiles/W3SVC627788864/ex100721.log		
282592	Wed Jul 21 2010 18:00:00	6363754	ma..	r/rrwxrwx	0	0	46777-128	C:/WINDOWS/system32/LogFiles/W3SVC1/ex100721.log		

Doing it my way.

Adding information to make it a "super" timeline.

- I like to add registry times and little else, adding too much information can make the timeline difficult to interpret. (Data Overload) However, there are times when more data is great.

- Adding registry MAC times is accomplished with the use of "regtime"

- Regtime.pl is a perl script written by Harlan Carvey and is included in the Sans Incident Response and Forensics Toolkit (SIFT)

- Log2timeline is a great utility written by Kristinn Gudjonsson for adding in Windows event logs, Dr. Watson logs, IIS and Apache logs and any number of other things. (The latest version contains 32 input modules.) In between writing this presentation and now delivering it, Kristinn has released a new version that makes my entire presentation obsolete. ***THANKS A LOT .*** We'll talk more about L2T in just a minute.

Doing it my way.

The process:

Extract the registry hives from %systemroot%/Windows/System32/config and the ntuser.dat files from their respective profiles in C:\Documents and Settings\<username> (Windows XP), C:\Users\<username> (Windows Vista and 7).

Perl regtime.pl –m <hivename> -r <filepath>

The <hivename> variables are:

HKLM/SAM, HKLM/SECURITY, HKLM/Software, HKLM/SYSTEM and HKEY_USER

Doing it my way

So.

Perl regtime.pl –m HKLM/SAM –r C:|Cases|registry|SAM >>fs_bodyfile

The double >> will append your new output to the end of the previous bodyfile. Using a single > will crush the data you already have in your bodyfile, I always make a copy of the fs-body first…just in case.

Repeat for each hive file and NTUSER.DAT you want to add…

Run mactime again.

Perl mactime.pl –d –b fs_bodyfile >super_timeline.csv

Doing it my way.

And there you have it. A human-readable timeline including registry keys in a sortable, searchable CSV file (easily opened in Excel).

The .csv format is also searchable via the UNIX grep command as well as many other stackable commands.

```
C:|tools|strings super_timeline.csv |grep –i inetmgr.exe
```

```
Sun Nov 16 2003 08:08:59,1301406,m..b,r/rrwxrwxrwx,0,0,47066-128-3,C:/WINDOWS/system32/inetmgr.exe
Mon Apr 23 2007 10:53:45,19456,m...,r/rrwxrwxrwx,0,0,37482-128-3,C:/WINDOWS/system32/inetsrv/inetmgr.exe
Mon Jun 11 2007 19:36:29,19456,...b,r/rrwxrwxrwx,0,0,37482-128-3,C:/WINDOWS/system32/inetsrv/inetmgr.exe
Tue Jun 12 2007 09:47:42,0,m...,0,0,0,0,HKLM/Software/Microsoft/Windows/CurrentVersion/App Paths/inetmgr.exe
Wed Nov 03 2010 12:01:34,19456,.a..,r/rrwxrwxrwx,0,0,37482-128-3,C:/WINDOWS/system32/inetsrv/inetmgr.exe
Wed Nov 03 2010 12:13:37,19456,..c.,r/rrwxrwxrwx,0,0,37482-128-3,C:/WINDOWS/system32/inetsrv/inetmgr.exe
Fri Jan 07 2011 04:32:59,1301406,..c.,r/rrwxrwxrwx,0,0,47066-128-3,C:/WINDOWS/system32/inetmgr.exe
Fri Jan 14 2011 08:02:01,1301406,.a..,r/rrwxrwxrwx,0,0,47066-128-3,C:/WINDOWS/system32/inetmgr.exe
```

Timestamp Alteration (Timestomping)

There are a couple of interesting things to notice on the previous slide.

- Inetmgr.exe has 2 separate locations on this server. The one in ~/inetsrv/ is a valid location. %systemroot%Windows/System32/ is not.
  - The binary in /System32 is malware.

- The date on that binary does not fit the rest of our system timeline, this server wasn't even put into production until 2007.
  - Sun Nov 16 2003 08:08:59,1301406,m..b,r/rrwxrwxrwx,0,0,47066-128-3,C:/WINDOWS/system32/inetmgr.exe
  - How do we find out if this has been altered or if it is just an anomaly?

Timestamp Alteration (Timestomping)

Some strong words from Vincent Liu, the creator of Timestomp, in an interview with CIO magazine May of 2007.

- "…*But forensic people don't know how good or bad their tools are, and they're going to court based on evidence gathered with those tools. You should test the validity of the tools you're using before you go to court. That's what we've done, and guess what? These tools can be fooled. We've proven that.*
  - **I agree, tools can be fooled. Good investigators cannot.**

- *"For any case that relies on digital forensic evidence, Liu says, It would be a cakewalk to come in and blow the case up. I can take any machine and make it look guilty, or not guilty. Whatever I want."*
  - **I beg to differ.**

Defeating Timestamp Alteration

Proving Mr. Liu wrong.

Remember that second set of attributes in the $MFT? $FILE_NAME?

- AS of the writing of this presentation there is nothing known that can change these attributes on a live system.
- How do we get at those attributes and make some sense out of the $MFT?
  - Harlan Carvey to the rescue (again)!

- Harlan has written a script called mft.pl that he recently made publicly available on www.regripper.net. This script parses out the data related to the MAC(B) attributes and outputs it in a (mostly) human-readable format.

Defeating Timestamp Alteration

Ripping the $MFT

- *Perl mft.pl c:|cases|registry|$MFT >ripped_mft.txt*
  - Simply outputs a .txt file containing the MAC(B) attributes stored in both $STANDARD_INFO and $FILE_NAME

- A little more grep-fu and we've got what we need.
  - *Strings ripped_mft.txt |grep –C 6 –i inetmgr.exe*
    - *-C 6 show 6 lines of context surrounding the search hit*
    - *-i ignore case*

Defeating Timestamp Alteration

```
--
   M: Sun Nov 16 15:08:59 2003 Z              Look familiar?
   A: Fri Jan 14 15:02:01 2011 Z
   C: Fri Jan  7 11:32:59 2011 Z
   B: Sun Nov 16 15:08:59 2003 Z
 0x0030 112  0   0x0000
0x0000
 FN: inetmgr.exe  Parent Ref: 2783  Parent Seq: 1       $F_N
   M: Wed Jul 21 17:40:21 2010 Z              Unaltered MAC(B)
   A: Wed Jul 21 17:40:21 2010 Z
   C: Wed Jul 21 17:40:21 2010 Z
   B: Wed Jul 21 17:40:21 2010 Z              Unaltered MAC(B)
 0x0080 72   1   0x0000
0x0000
```

Defeating Timestamp Alteration

Interpreting the output of our search.

The result of the 6 lines before and after the filename “inetmgr” are the MAC(b) times from $STANDARD_INFO showing the suspect Modifed and Birth times and the MAC(b) times from the $FILE_NAME attribute which shows the accurate birth date and time of the file.

- Definitive evidence of time alteration!
- The second set of timestamps blows away the use of timestomp.
- Trustwave is encountering the use of this regularly in our investigations.

My way just got easier

The latest version of log2timeline has added functionality that automates virtually everything I just talked about.

c:\perl\log2timeline\log2timeline.pl -r <directory> -z <timezone> -f <logtype> -w <outfile.csv>

- Version .60 runs on Windows as well as Linux
- It will automatically parse the registry hives, event logs, the $MFT, IIS logs and just about anything else you can think of.
- Nice work Kristinn, soon nobody will need me at all.

Demo Time

- **Create a super timeline from a postmortem image using fls, regtime and mactime.**

- **Perform a string search for known malicious files in that timeline.**

- **Verify timestamps of recovered files by comparing $SI and $FN from the $MFT**

Tools

Tools used and credit to their creators:

Regtime.pl and mft.pl

Harlan Carvey. Author of "Windows Forensic Analysis", "Windows Registry Forensics" and the supporting "Windows Incident Response" blog.

http://windowsir.blogspot.com

Fls and mactime.pl

Brian Carrier. Author of "Filesystem Forensic Analysis" and the creator of "The Sleuth Kit"

http://www.sleuthkit.org

Log2timeline

Kristinn Gudjonsson. Author GCFA Gold paper "Mastering the Super Timeline With log2timeline" and the forensics blog "IR and Forensic talk"

http://blog.kiddaland.net

Tools

More tools used and credit to their creators.

- The original mac_daddy was written by Rob Lee and was absorbed into mactime by Brian Carrier. It's only fair to mention Rob since he is The <u>original</u> MAC(b) daddy.

- Special thanks to Chris Pogue (@cpbeefcake) for teaching me his "Sniper Forensics" methodology and always having the time to mentor me, even when he doesn't have the time to mentor me.

Conclusion

Real world anti-forensics.

In the course of real world investigations we regularly encounter malware and attackers that are using anti-forensics techniques in an attempt to obfuscate their trail. As attackers start to use these tools more regularly, the investigators that work the cases need to be better armed and prepared to recognize and defeat their methods.

- Everything leaves trace evidence somewhere, so know how your tools work and not just that they do. (Locard's Exchange Principle)
- Remember that the tools do not make the investigator, it's the investigators use of the tools that make them effective.
- Be aware that these tactics are in use.

Trustwave®

SpiderLabs℠

Questions??

Trustwave®

SpiderLabs℠

Thank You!

Glenik@trustwave.com

http://eyeonforensics.blogspot.com/

@handlefree