Forensic Analysis of Spoliation and Other Discovery Violations Part I of a 2 Part Series: Macintosh Examinations

## **By Steve Bunting**

When the police investigate a crime and they execute a search warrant for digital evidence, the charged party usually isn't aware that the police are coming with warrant in hand. In essence, the search of the digital media is often achieved by surprise and the suspect has little or no time to dispose of evidence. Even if the defendant had some prior warning and subsequently deleted or secreted digital evidence, from a practical sense, there's no crime or penalty for doing so. Furthermore, the criminal defendant enjoys the right not to self-incriminate.

In matters involving litigation, the rules are much different. Once a party to potential litigation becomes aware of the reasonable possibility or likelihood of litigation, a duty attaches to all parties to preserve all potential evidence, including digital evidence. Even counsel for both parties has an obligation to instruct their clients and to ensure that preservation of evidence. In the Zubulake case<sup>1</sup>, the court stated that counsel has an affirmative duty to monitor their client's compliance with evidence preservation obligations. Thus obligated, the parties carefully preserve all evidence and provide that evidence to the other party when the discovery process begins, or at least, that's how it is supposed to work in an ideal world when everyone follows the rules.





Understandably, it must be a tough pill to swallow for a potential litigant to look at their digital media, know it contains information that harms their position, and then preserve it so it can be handed over to the opposing party to be used against them. And so the thought crosses their mind that it would be much better to either exclude that media from discovery or to destroy or alter the incriminating portions of the media so it can be safely turned over during discovery. Spoliation of evidence is the intentional, reckless, or negligent withholding, hiding, altering, fabricating, or destroying of evidence relevant to a legal proceeding. Thus withholding, deleting, or hiding evidence are forms of spoliation. More specifically, referencing Black's Law Dictionary in its ruling, an Arkansas court defined spoliation as "the intentional destruction of evidence and when established, [the] fact finder may draw [an] inference that [the] evidence destroyed was unfavorable to [the] party responsible for its spoliation."<sup>2</sup> Thus spoliation carries with it a very specific penalty in that the aggrieved party may legally infer the destroyed evidence was unfavorable, which often has a devastating impact on the party who destroyed the evidence.

Once the litigant heads down this path, the slope becomes treacherous and slippery. In addition to spoliation of evidence, often they are signing sworn declarations attesting to accuracy and completeness of the discovery materials, which in turn can form the basis for perjury and the case can quickly evolve into a criminal matter. Of course, spoliation requires proof, but once that proof is forensically established, things start to snowball and the penalties are usually worse than the outcome would have been otherwise, if the rules had been followed, which is, of course, the intent of the law and the rules in the first place. So let's turn our attention now to various forms of proof that can be used to establish spoliation. Part 1, this part, will discuss the artifacts found on the OS X or Macintosh operating system. Certainly, we can't cover all facets of such an examination, but will discuss some of the more common artifacts of interest in a spoliation case.

Before we get technical, let's look at the matter in a physical world. Let's assume that we are walking through snow covered terrain and we wish to hide our tracks. The simple way would be to cut an evergreen bough and use it to whisk away and obliterate those tracks. From a practical aspect, the whisking away can leave a pattern that is observable. The tracks may be gone, but the trace artifacts left by the branch are present. When we reach the end of our trail and are complete, we have eliminated the tracks, but are still left holding the branch in our hands. We can toss it or hide it, perhaps, but it can still likely be found in part or in whole. And let's not forget that somewhere there is evidence of where we cut this evergreen bough. This creates an interesting physical scenario and helps us view the process in the digital world.

52



In the digital world, the process is somewhat similar. Let's say the party uses an OS X built-in tool to wipe those tracks, which is "Erase Free Space", found in Disk Utility. This tool will zero out or wipe data in the free spaces of the selected media. Such an act, done after the duty to preserve attaches, is clearly an act intended to destroy potential evidence and certainly fits the definition of spoliation. Figure 1, below, shows an area in the unallocated clusters that contains data, or footprints in the snow, before any wiping has occurred. The data, or tracks in the snow, are plain to see. After the data has been wiped, as shown below in Figure 2, the data is gone, as one would expect. It is important to note, though, that in its place is a pattern of zeros, all zeros in fact. This pattern is analogous to the pattern left by the evergreen bough in our physical example.

Normally, one expects to see considerable data in the unallocated spaces in a normally functioning computer system. Files are continually being deleted, both by the system and by the user. When files are deleted, only the pointers to that data are changed. The data is not deleted when a file is deleted. The space occupying the data is marked as

								13	U	Inall	ocat	ed (	lust	ers																				
alda	[7:0]	Dan		A	Terr		<	 		De	da				E	] т		rint	IE	D:	-+				ala	all	- 51	- Eve			a n			
eius		rep	on		TEX			TEX		De	coue			OC .	1	1110	msc	npr	-	PI	ciun	-		.ons	oie	00	- FII	e exi	ents	· 12	t P	enn	ISSIC	ms
ptions	A	Co	depa	ge	• 1	₹,T	ext S	Style		品	Find	-					Cor	mpre	esse	d Vie	ew													
2454	8 0 0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	F4	B1	EB			13
2458	2 EA	54	EC	D4	Ξ4	5F	F1	3A	E5		FO		ES		EF	02	Ξ2		FO	59	E2	12	EE	C4	Ξ2		EC	CA	Ξ3		88	BC	E1	AD
2461	6 E 4	1F	DC		E1	DB		47		A5			DE		D9	CD		6D			DE	82		EA		A6	D7		DE	FC		CC	E2	72
2465	0 DD	42	E1	56		37				F3		CA							DC	47		82		90			D9	1E	D6	31		B7		01
2468	4 D9	20	D4	54	D9	57	D3	Ε7	D8	OF	D1	24	D6	3E	CF	EC	De	EC	D1	F5	D9	BE	DO	BE	D7	45	CE	28	D4	3D	CD	31	D7	36
2471	8 CE	EE	DB	59	D2	FO	D6	B6	D3	6E	D7	01	D2	25	D6	3C	D4	3F	D4	FE	DS	0C	D6	30	D5	BD	D7	2.5	DA.	4D	DS	FE	D6	B6
2475	2 05	10	D5	4A.	De	3A	D8	CA	D5	4F	D8	00	D1	68	De	26	DO	EE	DE	CC	D1	82	D4	FF	CF	79	D1	48	CF	04	D1	D9	CD	11
2478	603	DG	CA	4A	C9	45	CC	08	CA	76	cc	A1	CA	18	CB	E6	CD	52	CE	40	DO	8D	CF	D6	C9	92	CF	AS	CA	A9	CF	69	CE	44
2482	0 03	97	CA	29	CE	3E	CA	<b>B1</b>	CE	B6	CE	43	CE	CB	CF	87	CF	2F	CE	EB	DO	OA	DO	83	CF	81	D1	7B	cc	cc	CD	13	CB	AE
2485	4 CE	22	CD	B4	D3	15	CB	59	DO	FB	C7	EB	CF	28	C9	CE	CE	44	CB	41	CE	OA	C7	A3	CF	BD	C9	DF	CD	8F	CS	65	СВ	A2
2488	8 C 5	03	CB	94	C8	57	CA	SA	C5	DB	CA	69	C3	<b>B</b> 6	C9	61	C1	79	C4	28	C2	33	C5	F6	C5	00	CA	4A	C3	06	C7	74	C4	46

Figure 1 - Data found, starting at offset 24,559, in the unallocated clusters, before any wiping.

LD.		<b>⊋</b> _R	ASH	E~1	н.	Ē	14		Un	allo	cate	d Cl	uste	rs																_		
	Ш				•	1		111																								
ields	144	Repo	ort	A	Tex	t [	10 H	lex	iú	De	cod	e [	<b>a</b> C	oc	in.b	Tra	ansc	ript	I	Pi	ctur	e (		Cons	ole	00	• File	e Ext	ents	s (	P	ermi
ptions	* * A	Co	depa	ge	• ]	A T	ext S	Style	+	舟	Find		11				Cor	npre	esse	d Vi	ew											
02451	2 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
02454	4 0 0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0.0	00		0.0	00	00		00	00	00		0.0	00	00	00
02457	6 0 0																															00
02460	8 0.0																															00
02464	0 0 0																												00			00
02467	2 0 0																															00
02470	4 0 0																															00
02473	6 0 0																															00
02476	8 0 0											00																				00
02480	0 0 0																													00	00	00
02483	2 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 2 - Same exact location in the unallocated clusters after secure erase has occurred.



available and can be overwritten, but normally, one finds considerable data in these spaces. To find mostly zeros in the unallocated spaces of an active computer system is suspicious.

Now that we've seen the pattern left by the tool, let's see if there's evidence of the tool itself and perhaps some incriminating metadata. After all, a branch was used to wipe the tracks in the snow, let's see if there's evidence of that branch.

On a Macintosh operating system, OS X, there's no need to purchase a tool, as an excellent one is available for free. It is located, as previously mentioned, in the Disk Utility. If there's one thing for certain with OS X, it is a logging beast, which is to say that since OS X is based on BSD Unix, robust logging is hereditary. Believe it or not, Disk Utility has its own log and since many spoliation activities (formatting, erasing, securely erasing, encrypting disks, etc.) are carried out within the Disk Utility toolset, this log is a goldmine when examining spoliation issues, as nearly all Disk Utility actions are recorded here.

There are many ways to examine this log, including most any text editor, as it is a pure text-based log. My preference is to use Console, which is the OS X native utility for viewing and searching logs. There is a Disk Utility log for each user, which establishes individual accountability on multi-user systems, assuming each user has and uses their own account. The log is located at: ~/Library/Logs/ DiskUtility.Log.

With this log open, it is often wise to peruse through it and observe the activity. This log often covers long periods of time as it exclusive to Disk Utility. For example, my machine is used almost daily when I'm in my lab and Disk Utility is used frequently. As of this writing, there is activity in this log covering a 2 ½ year span.

In particular, you want to observe carefully the timeline the moment the duty to preserve attaches, or thereabouts. This is often referred to as the OS moment (Oh Shoot moment), when the party realizes they are in trouble and the urge to circle the wagons occurs. If suddenly, at that juncture, you see a lot of drives being formatted, securely wiped, and so forth, you have hit spoliation pay dirt.

To see if there was any erasing or erasing of free space (two different functions), filtering in Console for the string "erase" is a good starting point. Figure 3, below, shows different types of erasing activity on the system under review.

• • •	DiskUtility.log	
MARNIN AND		Q~erase Ø
Hide Log List Clear Display	Reload Ignore Sender Insert Marker Inspector	Filter
Hockage task     Adobe/Download     adobegc.log     Adobe/Download     adobegc.log     Adobe/Download     adobegc.log     Adobe/DCBroker.log     Adobe/PCBroker.log     Adobe/PCBroker.log     amt3.log     PApple TV Updater Logs     Pom.apple.Adote/Notes     Pom.apple.Notes     Pom.apple.Adote/Notes     Pom.apple.Notes     Pom.apple.Not	2015-03-24 10:18:18 -0400: 1 Pass Erase Free 2015-03-24 10:41:10 -0400: Secure Erase Free Sp in 22 minutes. 2015-03-24 10:41:10 -0400: Erase complete. canBeRestoreTarget No, with erase Yes canBeRestoreTarget No, with erase tree colfo-08-19 08:49:05 -0400: 1 Pass Erase Free Sp 2016-08-19 09:17:23 -0400: Secure Erase Tree Sp 2016-08-19 09:17:23 -0400: Secure Erase Tree Sp 2016-08-19 09:17:23 -0400: Secure Frase Tree Sp	Space bace completed successfully : "OLDSTUFF" : "OLDSTUFF" erased Erase Free Space : Space bace failed with the error:
	Size: 267 KB (Showing last 267 KB)	Earlier Later Now

Figure 3 – Using Console, DiskUtility.log file filtered for 'erase".



To distinguish between the two different erasing activities, if one erases a partition, they are replacing it with another partition. In that case, the security options default to "fast" or no wiping with zeros. In this case, data still can be recovered from that media. If one erases free space (Secure Erase Free Space), then all data in the free spaces are replaced with zeros and that data is, for all practical purposes, gone. Regardless, either can be evidence of spoliation and are captured in this log.

Another good review item for this log is to filter on the string "format" or "formatting". See Figure 4, below. You will then see all volumes that have been formatted with Disk Utility. You may see drives formatted shortly after preservation has attached, which can establish spoliation. You may also see drives formatted that have been withheld from discovery and by their very names appear relevant. For example, spoliation is all too common in digital rights litigation, where a defendant is accused of illegally downloading copyrighted materials.

If a drive, in such a case, were discovered in this log to have been formatted and named "MyMovieLibrary", such would be most significant where a device by that name had not been disclosed in discovery. Figure 4 shows a log filtered for "formatting". In it, one entry of particular import has been circled in red, which indicates a volume has been formatted and named "TimeMachineMacPro4TB". Such a discovery is

1997/36	C 🖓 👌 👌 🚺
lide Log List Clear Display F	Reload Ignore Sender Insert Marker Inspector Filter
▶ AdobeDownload	2015-07-09 18:13:41 -0400: Formatting disk13s2 as Mac OS Extended
adobegc.log	(Journaled) with name HFSPLUS
AdobelPCBroker.log	2015-07-27 12:00:59 -0400: Formatting disk12s2 as Mac OS Extended
AdobelPCBrokerCustomHook.l	(Journaled) with name 2TB SOFTWARE
Advanced Mac Cleaner.log	2015-08-29 15:51:04 -0400: Formatting disk15s2 as Mac OS Extended
amt3.log	(Journaled) with name OSXIRTOOLS
Apple TV Updater Logs	2015-08-29 18:09:38 -0400: Formatting disk15s2 as Windows NT Filesystem
CloudServices	with name MacClassImages
▶ com.apple.accountsd	Formatting Volume
com.apple.Notes	2015-10-01 09:50:44 -0400: Formatting disk14s2 as Mac OS Extended
▶ com.apple.NotesMigratorServi	(Journaled) with name ELCAPINSTALLER
▶ com.citrixonline.GoToMeeting	2016-01-06 17:38:02 -0500: Formatting
▶ com.citrixonline.WebDeployment	2016-01-18 14:00:26 -0500: Formatting disk14s2 as Mac OS Extended
CoreSimulator	(Journaled) with name MacProTimeMachine
▶ CoreSync	2016-01-18 14:15:54 -0500: Formatting disk14s2 as Mac OS Extended
▶ CrashReporter	(Journaled) with name TimeMachineMacPro4TB
▶ CreativeCloud	2016-08-18 13:52:37 -0400: Formatting disk15s1 as MS-DOS (FAT) with name
▶ CSXS	MYSTUFF
DiagnosticReports	2016-08-18 13:53:07 -0400: Formatting disk15s1 as MS-DOS (FAT) with name
DiscRecording.log	OLDSTUFF
DiskUtility.log	

Figure 4 - DiskUtility.log file viewed in Console and filtered for "formatting".

incredibly important in every case, but even more so when spoliation is the issue. If a party is trying to hide evidence, it is not likely they have disclosed their TimeMachine or backup drive, which provides a backup of nearly all of their data. How far it goes back will vary but, from this log, you have just discovered its existence and the spoliation case has become much stronger. Naturally, you'll need to seek production of this drive from the party.

Whether you've discovered a volume named Time Machine in the Disk Utility log or not, you'll still want to see if Time Machine is running, when it last backed up, and the name of the volume to which it is writing. This information is contained in the file / var/log/system.log. Again, you can view it in the application Console. To view Time Machine information, filter for the string "backupd" (backup daemon). If Time Machine is running, you'll find entries regardless of whether the backup disk is mounted or not. If mounted, you see references to it as shown in Figure 5 below (see volume name highlighted in blue). If not, it will mention the volume by name that couldn't be found. Either way, you will



know of its existence. If it exists and has not been disclosed, your spoliation case is again supported by this finding.

If you have a Time Machine drive, produced initially or produced later as a result of your discovering its existence, you'll want to compare what is currently on the party's machine with content of the Time Machine drive between the "OS moment" and when the Macintosh computer system was produced. It is during this period when the data starts to "disappear".

It may be that the party claims they don't have the Time Machine drive for whatever reason. That's an issue for the court, but that doesn't necessarily stop your examination of Time Machine. Time Machine has an obscure feature about which little is mentioned in forensic circles. I was doing some testing with Scott Pearson in April 2013 in Manila when we encountered a hidden file in the root named ".MobileBackups" coupled with a mounted volume by the same name. All of this is hidden from the regular user. Upon exploring this a little more, we discovered that whenever Time Machine can't backup to a drive that is not present, it maintains the Time Machine function by writing temporary Time Machine data to this hidden file. It is stored in the same format as a regular Time Machine drive, as shown in Figures 6 and 7.

Time Machine makes extensive use of link files. Where a file hasn't changed, there's a link file

• • •		system.log	
	Ç 📄 🍸 🚺		Q backupd
Hide Log List Clear Display	Reload Ignore Sender Insert Marker Inspector		Filter
com.apple.WebKit.Plugin.64_2 Mail_2016-08-12-095733_Mac suggestd_2016-08-04-08225 suggestd_2016-08-04-08480 ▶ System Diagnostic Reports FILES system.log ▼~/Library/Logs ▶ 11111 ACC 2016-01-28 07-39-39.log ACC.log ▶ accountsd ▶ Adobe ▶ Adobe CCX Welcome ▶ Adobe Creative Cloud Libraries ▶ Adobe Creative Cloud Libraries ▶ AdobeDownload adobegc.log AdobelPCBroker.log AdobelPCBroker.log AdobelPCBroker.log amt3.log ▶ Apple TV Updater Logs ▶ CloudServices ▶ com.apple.accountsd ▶ com.apple.Notes	Aug 19 10:00:44 Mac-Pro7516 c Backups.backupdb/Mac Pro7516/ Aug 19 10:00:44 Mac-Pro7516 c backups removed Aug 19 10:00:44 Mac-Pro7516 c Aug 19 11:01:14 Mac-Pro7516 c Aug 19 11:01:14 Mac-Pro7516 c TimeMachineMacPro4TB/Backups. Aug 19 11:01:16 Mac-Pro7516 c Aug 19 11:01:16 Mac-Pro7516 c Aug 19 11:01:16 Mac-Pro7516 c 2.86 TB available Aug 19 11:03:44 Mac-Pro7516 c Macintosh HD. Linked 9485. Aug 19 11:03:47 Mac-Pro7516 c Aug 19 11:03:48 Mac-Pro7516 c Aug 19 11:03:48 Mac-Pro7516 c Aug 19 11:03:48 Mac-Pro7516 c Volumes/TimeMachineMacPro4TB/ "Unable to get Recovery disk internal lookup" UserInfo={NS -69808}: Some information was Aug 19 11:05:36 Mac-Pro7516 c Backups.backupdb/Mac Pro7516 c Backups removed Aug 19 11:05:36 Mac-Pro7516 c	<pre>com.apple.backupd[68318]: Deleted /Volumes/T 2016-08-18-092729 (85.1 MB) com.apple.backupd[68318]: Post-backup thinni com.apple.backupd[68318]: Backup completed s com.apple.backupd[68500]: Starting automatic com.apple.backupd[68500]: Backing up to /dev backupdb com.apple.backupd[68500]: Will copy (145.8 M com.apple.backupd[68500]: Found 403 files (1 com.apple.backupd[68500]: Found 403 files (1 com.apple.backupd[68500]: Copied 619 items ( com.apple.backupd[68500]: Copied 619 items ( com.apple.backupd[68500]: Created new backup Backups.backupd[68500]: Created new backup Backups.backupd[68500]: Could not back up Backups.backupd[68500]: Could not back up Backups.backupd[68500]: Some information was un clocalizedDescription=Unable to get Recovery unavailable during an internal lookup} com.apple.backupd[68500]: Starting post-back com.apple.backupd[68500]: Deleted /Volumes/T 2016-08-18-101937 (96.2 MB) com.apple.backupd[68500]: Post-backup thinni com.apple.backupd[68500]: Backup completed s</pre>	<pre>imeMachineMacPro4TB/ ng complete: 1 expired uccessfully. backup /disk6s2: /Volumes/ B) from Macintosh HD 45.8 MB) needing backup including padding), 145.3 MB) from volume : 2016-08-19-110346 OS X Recovery to / Domain Code=-69808 available during an disk ref (error up thinning imeMachineMacPro4TB/ ng complete: 1 expired uccessfully.</pre>
	Size: 1	.1 MB	▲ Earlier V Later Now

Figure 5 - /var/log/system.log filtered for string 'backupd' to reveal Time Machine activity



Pictures	Name	Date Modified	Date Created
	DS_Store	Jul 2, 2016, 4:23 PM	Jul 1, 2016, 9:43 PM
Public	🔜 Macintosh HD	Aug 20, 2016, 8:16 PM	Aug 20, 2016, 8:16 PM
~	🔻 🖳 MobileBackups	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Saved Searches	🔻 📄 Backups.backupdb	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	lionserver	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Scans	2016-08-20-141701	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	Macintosh HD	Aug 20, 2016, 8:17 PM	Jan 1, 2001, 1:00 AM
Screenshots	v 📃 2016-08-21-145321	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	Macintosh HD	Yesterday, 8:53 PM	Jan 1, 2001, 1:00 AM
sbunting	2016-08-22-035353	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Services	🕨 📄 Macintosh HD	Today, 9:53 AM	Jan 1, 2001, 1:00 AM
Services	2016-08-22-035357	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Topaz Labs	2016-08-22-045429	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	2016-08-22-055501	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Trash	2016-08-22-065533	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	2016-08-22-065534	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Utilities	2016-08-22-075607	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	2016-08-22-085639	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Virtual Machines	2016-08-22-085640	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
	▶ 📃 2016-08-22-095713	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
Volumes	▶ 📃 2016-08-22-105745	Today, 10:03 PM	Jan 1, 2001, 1:00 AM
		T-J 40.00 DH	1 4 0004 4000 414

Figure 6 - Hidden Time Machine file in the root that is mounted and appears in /Volumes. It uses the same format as a regular Time Machine drive.

1	Magintoch HD	🔻 📄 201	6-0	8-20-141701	Today, 10:03 PM	Jan 1, 2001, 1:00 AM		Folder
	Macintoan nD	▼ 📃	Mag	intosh HD	Aug 20, 2016, 8:17 PM	Jan 1, 2001, 1:00 AM		Folder
i On	Disturss				Jun 23, 2016, 8:24 PM	Jan 1, 2001, 1:00 AM	263 bytes	TextEditocur
	Pictures			.com.apple.timemachine.donotpresent	May 13, 2015, 8:35 AM	Jan 1, 2001, 1:00 AM	Zero bytes	Document
<b>F</b>	D. L.F.	►		.DocumentRevisions-V100	Aug 20, 2016, 9:12 PM	Jan 1, 2001, 1:00 AM		Folder
	Public			.DS_Store	Aug 20, 2016, 2:44 PM	Jan 1, 2001, 1:00 AM	14 KB	TextEditocur
				.file	Feb 25, 2016, 11:50 PM	Jan 1, 2001, 1:00 AM	Zero bytes	TextEditocur
	Saved Searches			.SymAVQSFile	Sep 25, 2007, 3:11 AM	Jan 1, 2001, 1:00 AM	4 KB	Unix executab
<u> </u>		►		.syncprox_tmp	Jul 27, 2016, 7:16 AM	Jan 1, 2001, 1:00 AM		Folder
	Scans	►		Trashes	Dec 14, 2015, 5:07 AM	Jan 1, 2001, 1:00 AM		Folder
				acrobat8pro-en_US	May 2, 2008, 11:14 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
	Screenshots	▶		Applications	Yesterday, 9:53 PM	Jan 1, 2001, 1:00 AM		Folder
		►		bin	Jul 29, 2016, 12:04 PM	Jan 1, 2001, 1:00 AM		Folder
	sbunting			bridge2	Mar 19, 2008, 6:28 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
				cameraraw4	May 2, 2008, 11:38 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
	Services	►		Developer	Jun 20, 2012, 11:45 PM	Jan 1, 2001, 1:00 AM		Folder
				devicecentral1	Mar 19, 2008, 6:28 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
<b>F</b>	Topaz Labs	►		efi	Jun 21, 2012, 7:20 PM	Jan 1, 2001, 1:00 AM		Folder
				estoolkit2	Mar 19, 2008, 6:28 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
<b>F</b>	Trash		7	etc	Jan 15, 2016, 8:28 PM	Jan 1, 2001, 1:00 AM	11 bytes	Alias
	indon.			flash9-en_US	Mar 19, 2008, 6:28 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
<b>F</b>	Litilities	►		Groups	Oct 31, 2012, 2:19 PM	Jan 1, 2001, 1:00 AM		Folder
	Oundes	►		Incompatible Software	Jan 15, 2016, 9:35 PM	Jan 1, 2001, 1:00 AM		Folder
-	Matural Marchines			Installer Log File	Aug 5, 2008, 10:31 PM	Jan 1, 2001, 1:00 AM	1.1 MB	SimpleTocun
	Virtual Machines			installer.failurerequests	Aug 23, 2015, 4:35 AM	Jan 1, 2001, 1:00 AM	313 bytes	Document
			0	libmp3lame.dylib	Mar 26, 2007, 7:12 AM	Jan 1, 2001, 1:00 AM	724 KB	Dynamic Libra
	Volumes	►		Library	Aug 20, 2016, 8:41 PM	Jan 1, 2001, 1:00 AM		Folder
<u> </u>		►		lost+found	Jan 18, 2015, 2:58 AM	Jan 1, 2001, 1:00 AM		Folder
	Google Drive 2	►		opt	Aug 1, 2012, 10:35 PM	Jan 1, 2001, 1:00 AM		Folder
				photoshop10-en_US	May 2, 2008, 11:15 PM	Jan 1, 2001, 1:00 AM	181 bytes	TextEditocur
	Creative Cloud Files	►		platform-tools	Nov 3, 2012, 2:00 AM	Jan 1, 2001, 1:00 AM		Folder
		►		private	Aug 20, 2016, 8:17 PM	Jan 1, 2001, 1:00 AM		Folder
Devices	5	►		Recycled	Feb 24, 2010, 9:08 PM	Jan 1, 2001, 1:00 AM		Folder
	lieneenuer	►		sbin	Jul 29, 2016, 12:04 PM	Jan 1, 2001, 1:00 AM		Folder
	lionserver		8	SCDD1 Passcode.txt	May 20, 2011, 4:21 AM	Jan 1, 2001, 1:00 AM	10 bytes	Plain Text
		►		Shared Items	Jun 20, 2012, 2:09 AM	Jan 1, 2001, 1:00 AM		Folder
Shared				Synchronize! Volume ID	Feb 10, 2010, 7:46 PM	Jan 1, 2001, 1:00 AM	48 bytes	Document
	Alex MacBook Pro	►		System	Jul 29, 2016, 12:04 PM	Jan 1, 2001, 1:00 AM		Folder
		►		TempForiPhoto	Jun 20, 2012, 2:56 AM	Jan 1, 2001, 1:00 AM		Folder
	MacBook Air de Paula		2	tmp	Jan 15, 2016, 8:28 PM	Jan 1, 2001, 1:00 AM	11 bytes	Alias
			7	User Guides And Information	Jun 16, 2009, 12:46 AM	Jan 1, 2001, 1:00 AM	60 bytes	Alias
	MacBook Pro de Laur	►		Users	Aug 20, 2016, 8:17 PM	Jan 1, 2001, 1:00 AM		Folder
	macbook rio de Laur	►		usr	Aug 20, 2016, 8:22 PM	Jan 1, 2001, 1:00 AM		Folder
	machaokaro_4c59		2	var	Jan 15, 2016, 8:28 PM	Jan 1, 2001, 1:00 AM	11 bytes	Alias
	macoookpro-4000	►		Virtual Machines	Today, 10:28 AM	Jan 1, 2001, 1:00 AM		Folder

Figure 7 - Inside each folder bearing a timestamp name, there's a complete directory structure of the entire drive.

pointing back in time to where it actually exists when it was last changed. It takes some getting used to. EnCase has an excellent parser for stepping through these files. The point here is not to make you an expert on examining Time Machine files, but instead

to point out their criticality in spoliation cases and to point out the existence of this hidden Time Machine, which can be a virtual gold mine in any spoliation case. Very few forensic examiners know about the hidden copy of Time Machine and even fewer users,



so likely it will not be touched by a user trying to hide his or her tracks. As with any Time Machine examination, you should compare what you find in this temporary Time Machine with the periods before and after the duty to preserve attaches, especially as production for discovery commences.

Another under-exploited resource for spoliation examinations is the Macintosh Quick Look Thumbnail Cache. This database and cache image storage supports the Quick Look function in Finder and provides the cached thumbnail images that you see when you open a folder in Finder. Thus the act of opening a folder creates a thumbnail of that file's content. With Windows, only images have cached thumbnails. Mac supports other formats and thus you can expect to see cached thumbnails of documents and images.

The path to this file is deep and obscure and buried below /private/var/folders.

The bottom level folder is named "com.apple.QuickLook.thumbnailcache".

Between these two, you will find randomly named GUID folders, so it is easiest to filter your forensic tool to locate the string :

"com.apple.QuickLook.thumbnailcache". Therein you will find a set of SQLite database files (index.sqlite index.sqlite-wal & index.sqllite.shm) along with the images themselves in the thumbnails.data file, as shown below in Figure 8.

When you review the index.sqlite database, make certain to copy out the write-ahead-log (wal) and shared memory (shm) files. Data is first written to the wal file and later committed. Were you to read the database file alone, you would miss data contained in the 'wal' file! The database contains a series of related tables. One table, shown in Figure 9, is 'files' and contains the path and file names for which cached thumbnails have been created.

The 'files' table is linked or related to the 'thumbnails' table, which is shown in Figure 10. For each entry in the 'files' table, this table tracks the 'last\_hit\_date' and hit counts. Further, this table

🔴 😑 📄 /private/var/folders/2y	/k3246zt93b1650_j91pgj0tw000	0gn/C/com.apple.Qu	ickLook.thumbnailca	che
				0 🗊 »
Back View	Arrange Share Edit Tags	Path New Folder	Quick Look Action	Get Info Delete
Applications	Name	Date Modified	Date Created	Size 🗸
Applications	thumbnails.data	7:39 PM	6:52 PM	125 KB
Applications	index.sqlite-wal	7:39 PM	6:52 PM	74 KB I
Applications	📄 index.sqlite	6:53 PM	6:52 PM	66 KB I
	index.sqlite-shm	6:53 PM	6:52 PM	33 KB
BDF Course	.DS_Store	6:46 PM	6:46 PM	6 KB
	thumbnails.fraghandler	7:39 PM	7:39 PM	425 bytes
com.apple.QuickLook.thumbn	exclusive	6:52 PM	6:52 PM	Zero bytes
complete				
CoreServices				
DB TY 2015				
	📓 Macintosh HD > 📄 > 📄 >	🖿 > 🖿 > 🖿 > 🖿 (	C > 📄 com.apple.Quick	Look.thumbnailcache
	7 items, 121.18 GB av	ailable		

Figure 8 - Contents of the Quick Look Thumbnail Cache folder



• • •		📄 i	index.sqlite			
Import Export		R	eload Vacuum		Clea	ar Log
Q Search		2	chema Data	SQL Log		
TABLES		folder		file_name	fs_id	versi
files	1	/Users/sbunting/Desktop		Unknown.jpg	/.file/id=657	Ø
pending_secure_delete_buffer	2	/Users/sbunting/Desktop		RIFFJTAG_UsersManual.pdf	/.file/id=657	0
preferences	3	/Users/sbunting/Desktop/Stockholm Photos fro	m Mike Chris Hess	DSC03980-1.jpg	/.file/id=657	O
reserved buffer	4	/Users/sbunting/Desktop		DSC04331.jpg	/.file/id=657	0
solite master	5	/Users/sbunting/Desktop/Stockholm Photos fro	m Mike Chris Hess	DSC03961.jpg	/.file/id=657	Ø
thumbnails	6	/Users/sbunting/Desktop		IMG_2306.m4v	/.file/id=657	O
	7	/Users/sbunting/Desktop/Stockholm Photos fro	m Mike Chris Hess	20160823_182307.jpg	/.file/id=657	O
OPTIONS	8	/Users/sbunting/Desktop/Stockholm Photos fro	m Mike Chris Hess	DSC03980.jpg	/.file/id=657	0
main	9	/Users/sbunting/Dropbox/01 Confidential matte	er	Screen Shot 2016-08-17 at 2.59.14 PM.png	/.file/id=657	O
	10	/Users/sbunting/Desktop/Stockholm Photos fro	m Mike Chris Hess	20160823_182305.jpg	/.file/id=657	O
			0		Add filte	er
					22 -6 22	
+ 💀 -	+			O	-33 01 33	

Figure 9 - The 'files' table is displayed in **Base**, revealing file names and paths for which thumbnail caches have been created.

						index.sqlite						
Import Export						Reload Vacu	um					Clear Log
Q Search						Schema	Data	SQL	Log			
TABLES		file_id	size	icon_mode	hit_count	last_hit_date	width	height	bitspercomponent	bitsperpixel	bytesperrow	bitmapinfo
files	24	24	16.0	1	1	493925991	16	16	8	32	64	1
pending_secure_delete_buffer	25	25	16.0	1	1	493925991	16	16	8	32	64	1
preferences	26	26	16.0	1	1	493925991	16	16	8	32	64	1
reserved_buffer	27	27	16.0	1	1	493925991	16	16	8	32	64	1
sqlite_master	28	28	16.0	1	1	493925991	16	16	8	32	64	1
thumbnails	29	29	16.0	1	1	493925991	16	16	8	32	64	1
OPTIONS	30	30	16.0	1	1	493925991	16	16	8	32	64	1
main	31	31	16.0	1	1	493925991	16	16	8	32	64	1
	32	32	16.0	1	1	493925991	16	16	8	32	64	1
	33	33	16.0	1	1	493925991	16	16	8	32	64	1
												Add filter
+ 0-	+										0-33 of 3	3

Figure 10 - The 'thumbnails' table is displayed, in **Base** in which the last hit date, hit count, and pointers to the thumbnail cache are contained.

contains the pointers to the actual thumb image in the thumbnails.data file.

While you can manually step through these entries, it is no fun and very time consuming. Simon Key, from Guidance Software, has written an excellent EnScript for EnCase 7 / 8 that parses this information and pulls the images out as well, as shown in Figure 11. Simon has also posted an explanation of how this feature works and of its forensic import. This blog can be found at: <u>http://encase-forensic-blog.guidancesoftware.com/2014/05/examination-of-mac-os-x-quick-look.html</u>

The import of this data to a spoliation examination is quite simple. You can see files that likely once existed and are no longer present. In the above image, Figure 11, if this were a digital rights case and the defendant claimed to have never downloaded the above song, imagine the impact of



Quick Look Metadata	
Source Folder	/Users/ /Music/iTunes/iTunes Media/Music/Avril Lavigne/Diamond Collection
Source FileName	02 Losing Grip.mp3
Source FsId	/.file/id=6571367.414842
ContentRect	{{3, 3}, {58, 58}}
IconMode	1
HitCount	1
LastHitDate(raw)	366,936,840
LastHitDate(Iso8601)	2012-08-17T22:54:002
LastHitDate	08/17/12 06:54:00 PM
Width	64
Height	64
BitsPerComponent	8
BitsPerpixel	32
BytesPerRow	256
BitmapInfo	1
BitmapDataLocation	805,120
BitmapDataLength	16,384
PlistBufferLocation	282,872
PlistBufferLength	104
Flavor	6
FileRowld	44
ThumbnailsRowld	49
Picture Diamond (2	

Figure 11 - EnCase 7 Quick Look Thumbnail Cache Parser results. All this Quick Look Metadata, including the image itself, is parsed and presented in this view. Note the file name, path, and last hit timestamps.

this thumbnail cache entry on that claim. The last\_hit\_date will often reflect when the user was last reviewing the files, which is often after preservation attaches and before production of discovery. File names are often indicative of their contents and can establish relevance, which is supported by a thumbnail cache created by its contents. When such relevant appearing files are no longer present and the timestamp points to that critical "OS moment", you have found evidence of potential spoliation. You also have a file name to search for its presence in other critical areas, including the Time Machine.

Finding deleted files on a Macintosh, aside from those found in Time Machine, is a challenging endeavor. When files are deleted on a Mac, unlike Windows NTFS or FAT file systems, the Catalog's B-Tree structure very efficiency removes file metadata shortly after a deletion. Such leaves carving as the primary means of recovering deleted files on a Macintosh system. There are EnScripts that parse the Journal file, from which some file recoveries can be made. When files are carved, sometimes they contain internal metadata that can establish identity and timestamps in addition to their contents, any of which could be used as evidence of spoliation. Such data can also be used in conjunction with other findings, such as Quick Look Thumbnail Cache, for example.

Going back to covering our tracks in the snow, we should look also for evidence of the evergreen branch, which is to say in a digital world that we are looking for programs that are used to clean, remove, or destroy data and artifacts. As tools or programs come and go, it is often a good practice to search



Google for something like "OS X Evidence Cleaner". In this manner, you'll see the most popular tools for removing evidence. You can fashion search strings to search the case for the presence of such tools, either installed or when the user was searching for such tools.

An excellent tool for processing Macintosh systems is Sumuri's Recon. Recon allows you to select processing modules or plugins for various types of examinations. Once selected, the modules are run and results returned in the "Result Viewer", as shown below in Figure 12 below. Figure 12 shows the Result Viewer with the list of all installed applications displayed. In this case, we have selected the evidence cleaning tool known as "CCleaner". It shows when it was installed. If such coincided with the critical "OS Moment", you are showing the user installed a tool to remove evidence at a time when evidence was supposed to be preserved.

While this view is important, the Advanced Analysis section contains more information concerning this tool, as shown in Figure 13. You can see the settings for this tool, which is to say, which artifacts or evidence it is configured to clean or remove. Since this tool was used, it is a good idea to install and test the tool that was used so that you can observe first

•••				RECON For MAC OS	6 X - Result Viewer [0	DS X]	
Case Info			Installed Appl	ications Globa	l Search	Global Timeline	Global Reports
Plugin Search	Keywo	rd Searcl	h	🗌 🗮 Time Line 🗌	Search C Sł	now All	HTML ᅌ Bookmarks ᅌ 📋 Report
System				Applicatio	ns		Detailed Information Detach Full
Bash History			Record No.	Name	Version	Date Added	Plugin: Installed Applications
Installed Applications	200		00266	Calculator	201	6/01/15 19:24:28	Category: Applications
Recent Items	201		00267	Calendar	201	6/01/15 19:24:28	System Time Zone: Europe/Stockholm GMT+
🗑 Trash	202		00268	calibre	201	6/01/15 19:35:35	User Selected Time Zone: Europe/Stockholm
W USB Attached	203		00270	CalorieKing Nutrition and	201	6/01/15 19:36:17	CE1-6M1+1.00
Advanced Analysis	204		00279	CameraWindow	201	15/12/14 11:12:44	Record No.: 00360
	205		00272	Canon iP90 Setup Utility	201	5/12/14 11:34:29	Preview Detach Full
	206		00357	Cardiris for ScanSnap	201	15/12/14 11:30:15	
Data Destruction	207		00358	Cathode	201	6/01/15 19:35:34	
* Data Destruction	208		00360	CCleaner	201	6/08/18 19:21:17	
	209		00362	CCleanerHelper	201	5/12/17 14:12:36	
	210		00363	CCleanerSystemMonitor	201	5/12/17 14:12:36	No Preview Available
	211		00853	CCLibrary	201	6/06/15 18:06:35	
	212		00854	CCXProcess	201	6/06/15 18:06:39	
	213		00801	CD Spin Doctor	201	15/12/14 11:06:12	
			00070	000011			



• •				RECON For MAC OS X - Result Viewer [OS	X]		
Case Info	CCleaner		CCleaner	Global Search	Global Timeline	Global Reports	
Plugin Search	Keywo	ord Search	h	C Sho	w All □ → Export H	TML 🗘 Bookmarks 🗘 📋 Report	
System				Preferences Status		Detailed Information Detach Full	
Bash History			Record No.	Key	Value	Plugin: CCleaner	
Sealed Applications	1		00001	CHROME_BROWSER_COOKIES_FEATURE_ID	YES	Category: Preferences	
Recent Items	2		00002	com.apple.trackpad.fourFingerHorizSwipeGesture	2	System Time Zone: Europe/Stockholm GMT+2	
W Trash	3		00003	FIREFOX_BROWSER_FORMS_AND_SEARCH_HIST	NO	User Selected Time Zone: Europe/Stockholm-	
JUSB Attached	4		00004	FIREFOX_BROWSER_CACHE_FEATURE_ID	YES	CE1-0M1+1.00	
Advanced Analysis	5		00005	Adobe Lightroom	NO	Record No.: 00007	
	6		00006	Google Drive	NO	Preview Detach Full	
CleanMyMag 2	7		00007	SecureDeletionMethod	Simple Overwrite (1 Pass)		
Data Destruction	8		00008	Finder	NO		
	9		00009	Cyberduck	NO		
	10		00010	Automator Runner	NO		
	11		00011	NSDocumentAutosaveOldDocumentDayThreshold	14	No Preview Available	
	12		00012	CleaningWarningMessageBox_DialogSupressed	NO		
	13		00013	CUSTOM_FILES_AND_FOLDERS_FEATURE_ID	NO		
	14		00014	SYSTEM_LOGS_FEATURE_ID	NO		
			00045		VEO		

Figure 13 - CCLeaner details are displayed in the Advanced Analysis section. There are 223 settings or properties that are parsed. Only 14 are seen here.

eForensics M a g a z i n e

> hand its behavior, default settings, and any patterns or temporary files it may leave behind.

> Figure 14, shown below, shows the USB devices that have been attached. These are tracked in the system log. You can find them manually by searching for the string USBMSC if you wish. Recon does a nice job of parsing and presenting them. You can sort by time in Recon and if you see USB devices attached during the period around the "OS moment", you should consider that data may have been moved elsewhere. You should seek to make those devices part of discovery if they are not already included, as you will

have the make, model, and serial number of the USB device by which to identify it.

The Trash Can, shown in Figure 15 below, is where deleted files are initially sent when deleted. They are not really deleted until the Trash Can is emptied. While one would think that someone trying to destroy evidence would do a good job, they just might forget to empty it. It has happened in the past, because some users are just not that computer savvy, thus one should always check the obvious places by checking the Trash Can. You might get lucky. If you do, it will also show the timestamp for when it was deleted or 'added to the Trash Can'.

•••				RECON For MAC O	S X - Result Viev	ver [OS X]	
Case Info			JSB Attached	Globa	al Search	Global Timeline	Global Reports
Plugin Search	Keywo	ord Search		Time Line	Search	C Show All □ → Export HT	ML ᅌ Bookmarks ᅌ 📋 Report
System				USB Attached List	HFS Device List		Detailed Information Detach Full
Bash History			Record No.	USB Serial Number	Vendor ID	Timestamp	Record No.: 00010
<ul> <li>Installed Applications</li> </ul>	1		00010	S25HNYAG300258B	0x4e8	Aug 26 20:51:15	USB Social Number: S25UNVAC2002598
Recent Items     Treat	2		00009	0x0000000	0x154b	Aug 26 18:15:05	USB Senar Number: S25HN1A03002568
IISR Attached	3		80000	AA011021140002500949	0x781	Aug 26 18:14:25	Vendor ID: 0x4e8
Advanced Analysis	4		00007	AA010301131813443878	0x781	Aug 26 18:13:29	Device Release Number: 0x1
Black Hole	5		00006	57583431413234323035	0x1058	Aug 26 16:11:20	
CCleaner	6		00005	0x00000000	0x8564	Aug 26 14:42:12	Preview Detach Full
CleanMyMac 2	7		00004	07083B719A560857	0x1b1c	Aug 26 14:05:22	
Data Destruction	8		00003	07083B719A560857	0x1b1c	Aug 26 11:21:05	
	9		00002	000002624127	0x64f	Aug 26 10:33:36	
	10		00001	07083B719A560857	0x1b1c	Aug 26 10:31:19	
	11		00024	07083B719A560857	0x1b1c	Aug 25 16:28:27	No Preview Available
	12		00023	20097064660261C4588	0x2009	Aug 25 16:06:15	
	13		00022	07083B719A560857	0x1b1c	Aug 25 16:04:27	
	14		00021	57583431413535484A53	0x1058	Aug 25 15:01:34	
	-			004500440040055	A .AA		



• • •				RECON For	RECON For MAC OS X - Result Viewer [OS X]					
Case Info	Trash			- Alexandre	Global Search			Global Timeline	Global Reports	
Plugin Search	Keywo	ord Sea	irch	🗌 🗮 Time Lir	ne 🔍	Search	C Show All	Export H	TML ᅌ Bookmarks ᅌ 📋 Report	
System Bash History					Items				Detailed Information Detach Full	
Sealed Applications	-		Record No.	File Name	File Exist	Suffix	File Size (KB)	Date Added	User Selected Time Zone: Europe/Stockholm- CET_GMT+1:00	
Recent Items	1		00001	exclusive	YES		00000000	2016/07/24 05:59:23		
🧊 Trash	2		00002	index 7.39.42 PM.sqlite	TES VEC	sqlite	00000064	2016/08/26 18:39:26	Record No.: 00003 System Account: shunting	
🖉 USB Attached	-		00003	index.sqiite	1ES	squite	00000204	2010/08/20 17:14:19	oʻjetenni essenni y	
Advanced Analysis	4		00004	index.sqlite-snm	TES	squite-s	00000032	2016/08/26 17:51:11	File Name: index.sqlite	
Black Hole	5		00005	index.sqiite-wai	YES	sqlite-wai	00000068	2016/08/26 17:49:44	Preview Detech Full	
CCleaner	6		00006	resetreason	YES		00000000	2016/08/25 00:10:55		
CleanMyMac 2	7		00007	thumbnails.data	YES	data	00003484	2016/08/26 17:15:59		
Data Destruction	8		00008	thumbnails.fraghandler	YES	fraghan	00000001	2016/08/26 17:15:44		
									No Preview Available	

Figure 15 - The Trash Can could contain folders or files that the user deleted and simply forget to empty the Trash Can. It does happen!



The OS X operating system, just like its cousin Windows, stores a vast collection of recent items. Nearly all are in "plist" or property list files, which serve much the same function in OS X as does the registry in Windows. The number of plist files that are parsed and represented in this view is quite impressive. Some of system plist files are retained by application-specific plist files. Regardless, they are all aggregated in the Recent Items view in Recon. Figure 16, below, shows Recent Hosts and Recent Servers to which the computer in question was connected over the network. The import in spoliation cases is that here you will find remote or networked computers that should be and have not been made part of discovery.

Under the Advanced Analysis section, you will find a category for Data Destruction. While we've already discussed it earlier, this module searches for strings in the Disk Utility log relating to data destruction, as shown in Figure 17 below.

When using automated tools, one can save time and methodically carry out a large number of specialized tasks. These tools can point to areas in need of more in-depth analysis. One must remember, however, that they are no substitute for a knowledgeable examiner. They assist the examiner only. The examiner must know the tool and its limitations. The

• •				RECON For MAC OS >	K - Result Viewer [OS X]		
Case Info		Rec	ent Items	Global S	Search Global Timeline	Global Reports	
Plugin Search	Keywo	ord Search		🗮 Time Line	Search C Show All	HTML ᅌ Bookmarks ᅌ 📋 Report	
System				Files		Detailed Information Detach Full	
Bash History		Record No.	Category	Name		Category: RecentHosts	
Installed Applications		00000	Descattlests	Mas Des 7540	amb.//Mack/00Dec7510_amb_tep.local	Name: Mac Pro7516	
Necent Items	658	00658	RecentHosts	Mac Pro/516	smb://Mac%20Pro/516smbtcp.local	Path: smb://mac%20Pro/516smbtcp.iocal	
🗑 Trash	659	00659	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local	Artifacts Source: /Users/sbunting/Library/	
🖉 USB Attached	660	00660	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local	Application Support/com.apple.sharedfilelist/	
Advanced Analysis	661	00661	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local	com.apple.LSSharedFileList.RecentHosts.sfl	
Black Hole	662	00662	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local	Browiew Dotach Full	
CCleaner	663	00663	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local	Preview Detach Full	
🚇 CleanMyMac 2	664	00664	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local		
Data Destruction	665	00665	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local		
	666	00666	RecentHosts	Mac Pro7516	smb://Mac%20Pro7516smbtcp.local		
	667	00667	RecentServers	6TBStripedRAID	/Volumes/6TBStripedRAID/	No Broview Available	
	668	00668	RecentServers	sbunting	/Volumes/sbunting/	No Preview Available	
	669	00669	RecentServers	Data6TB	/Volumes/Data6TB/		
	670	00670	RecentServers	2TB SOFTWARE	/Volumes/2TB SOFTWARE/		
	671	00671	RecentServers	Data	/Volumes/Data/		



• • •	RECON For MAC OS X - Result Viewer [OS X]	
Case Info	Clobal Search	Global Reports
Plugin Search	Keyword Search	ML ᅌ Bookmarks ᅌ 📋 Report
System Bash History Advanced Analysis	Secure Erase         Command Line         Recent Applications         Cleaner Applications           Record No.         Title         Status         Timestamp           1         00001         RECON         2015/12/17 08:53:00	Detailed Information         Detach         Full           Plugin: Data Destruction         Category: Secure Erase         Full
Data Destruction	2 00002 Untitled 1 Stopped 2015/12/18 11:20:49	System Time Zone: Europe/Stockholm GMT+2 User Selected Time Zone: Europe/Stockholm- Preview Detach Full No Preview Available

Figure 17 - Data Destruction looks for strings in the Disk Utility log pertaining to secure wiping, etc.



examiner must fill in the gaps; otherwise, important data can be missed. The Data Destruction module tests the Disk Utility log for activity carried out by the GUI (Graphical User Interface) known as the Disk Utility. The GUI isn't the only means of data destruction built into OS X. Many commands issued in Terminal can be used to delete or destroy data.

For example, prior to the El Capitan release of OS X, there used to be a feature to allow the Trash Can to do a secure erase. Due to a limitation in that feature, Apple could not guarantee that this feature could do what it appeared to do, which is to guarantee a file or folder is securely erased. Accordingly, that feature is not available in El Capitan. Despite such, one can effectively carry out the same task in Terminal, by issuing the "srm" command, which is the secure version of the "rm" command, which means to remove or delete a file or folder. The 'srm' command, in its default mode, will make 35 passes as it overwrites the data. That is DOD-grade wiping on steroids, so don't doubt for one minute its effectiveness in destroying data. So where does one find evidence of commands used in Terminal? The answer lies in the history file, which exists for each user. To view this file in Recon, look to the Bash History view, as shown in Figure 18 below. By searching for 'rm' we find, in addition to some extraneous hits, one "rm" and one "srm" command in this case.

In OS X, there are no timestamps for each action in the history file, however, since the commands appear in the sequence executed, one can sometimes impute a range when something occurred by examining other items before and after the command in question. Sometimes you can see a command and find the results of that command and obtain timestamps in that manner.

Let's consider one more Terminal command that destroys data. Recall earlier we discussed how to "Erase Free Space" using the GUI, Disk Utility. If that occurred using the GUI, we would expect an entry in the DiskUtility.log file. Starting with the El Capitan release of OS X, Disk Utility was stripped of much of

		RECON For MAC OS X - Result Viewer [	IX 2C	
Case Info	Bash History	Global Search	Global Timeline	Global Reports
Plugin Search		C St	now All C+ Export HT	ML ᅌ Bookmarks ᅌ 📋 Report
System       Bash History       Advanced Analysis       Data Destruction   6 7	Record No. 00368 s 00498 s	Command Line sudo rm -rf /opt/local/var/macports/build/* srm /Users/sbunting/Desktop/untitled\ text.txt	Command	Detailed Information     Detach     Full       Preview     Detach     Full   No Preview Available

Figure 18 - An 'rm' and a 'srm' command found in the Terminal bash history file.



its former functionality. To wipe free space now, a terminal command will get the job done. That command is diskutil secureErase freespace LEVEL /Volumes/DRIVE, where DRIVE is replaced with the name of the mounted volume and LEVEL is replaced with a number from 0 to 4, which means:

- 0 writes zeroes to the disk once
- 1 writes a series of random numbers
- 2 writes zeroes 7 times
- 3 writes zeroes 35 times
- 4 writes zeroes 3 times

This command would also appear in the history file and one should search for the string "secureErase" to locate this activity. The command "diskutil" is followed by many 'verbs' that are associated with various data destroying activities. Below is a list of strings for which you should search in the history file for other forms of possible spoliation:

- eraseDisk
- eraseVolume
- reformat
- eraseOptical
- zeroDisk

- randomDisk
- secureErase
- partitionDisk

Alternatively, you may just want to search for "disktutil", which will return all activity involving "diskutil", which is shown below in Figure 19. You'll have, perhaps, more data to review, but you will be more thorough with that methodology.

Conducting a spoliation examination is a specialized exam for sure. The focus is on the various means of data destruction that are built into the operating system and also third-party tools. In addition, you are also making many before and after comparisons to show that data once present was deleted. As you have seen, the tools can be found as well as the patterns and artifacts they leave behind. It is very difficult to interact with a computer without leaving behind trace evidence. Spoliation is no different. There is usually plenty of evidence for the skilled examiner to establish spoliation when it occurs. This is not a complete treatise on spoliation, nor could it be. It is, however, a road map to follow that will uncover most of the more common spoliation activities. As always, one must be alert to the unusual and to always be inquisitive.

```
Mac-Pro7516:~ sbunting$ history |grep -i diskutil
510 sudo diskutil secureErase freespace 1 /Volumes/NO\ NAME
516 history |grep -i diskutil
Mac-Pro7516:~ sbunting$
```

Figure 19 - From the OS X terminal interface, history is shown, but piped to a grep command that is filtering for the string 'diskutil'. This returns all entries with 'diskutil'.



Part 2 of this 2-part series will focus on spoliation evidence found on the Windows operating system. The principles are largely the same, but the two systems are very different and hence the artifacts left behind.

## References:

- Zubulake v. UBS Warburg, LLC, 229 F.R.D. 422 (S.D.N.Y. 2004)
- 342 Ark at 146, 27 S.W.3d at 388 (quoting BLACK'S LAW DICTIONARY 1401 (6th ed. 1990) (content in brackets inserted by the court)).



## About the author: Steve Bunting

Steve Bunting, the author, is one of the pioneers in the field of digital forensics with over 17 years in the field. He spent ten of those years in digital forensics during his 35-year law enforcement career and seven of those years in support of the private sector.

He has been a presenter at several seminars and workshops, the author of numerous "white papers", the principal author of <u>EnCase Computer</u> Forensics - The Official EnCE: EnCase Certified Examiner Study Guide, 3<sup>rd</sup> Edition, the co-author of <u>Mastering Windows Network Forensics and</u> Investigation, the author of <u>EnCase Computer Forensics</u>—The Official EnCE: EnCase Certified Examiner Study Guide, 2<sup>nd</sup> Edition, the co-author of Mastering Windows Network Forensics and Investigation 2<sup>nd</sup> Edition, the author of <u>EnCase Computer Forensics</u>—The Official EnCE: EnCase Certified Examiner Study Guide, 3<sup>rd</sup> Edition (all published by Wiley).