JOURNAL OF FORENSIC SCIENCES

Forensic Investigation of Cooperative Storage Cloud Service: Symform as a Case Study

Journal:	*Journal of Forensic Sciences*
Manuscript ID	JOFS-15-646.R1
Manuscript Type:	Paper
Date Submitted by the Author:	17-May-2016
Complete List of Authors:	Teing, Yee-Yang; University of Salford, School of Computing, Science and Engineering Dehghantanha, Ali; University of Salford, School of Computing, Science and Engineering Choo, Kim-Kwang Raymond; University of South Australia, Information Assurance Research Group, Advanced Computing Research Centre Dargahi, Tooska; West Tehran Branch, Azad University, Computer Engineering Conti, Mauro; University of Padua, Department of Mathematics
Keywords:	Forensic Science, Cloud Forensics, Digital Forensics, Mobile App Forensics, Mobile Device Forensics, Peer-to-Peer Forensics, Cooperative Cloud, Symform Analysis

Forensic Investigation of Cooperative Storage Cloud Service: Symform as a Case Study

Yee-Yang Teing,[1] B.Sc.; Ali Dehghantanha,[1] Ph.D.; Kim-Kwang Raymond Choo,[2.3] Ph.D.; Tooska Dargahi,[4] Ph.D.; and Mauro Conti,[5] Ph.D.

[1]School of Computing, Science and Engineering, University of Salford, Manchester, United Kingdom.

[2]Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX.

[3]Information Assurance Research Group, University of South Australia, Mawson Lakes, Australia.

[4]Department of Computer Engineering, West Tehran Branch, Islamic Azad University, Tehran, Iran.

[5]Department of Mathematics, University of Padua, Padua, Italy.

ABSTRACT: Researchers envisioned Storage as a Service (StaaS) as an effective solution to the distributed management of digital data. Cooperative storage cloud forensic is relatively new and is an under-explored area of research. Using Symform as a case study, we seek to determine the data remnants from the use of cooperative cloud storage services. In particular, we consider both mobile devices and personal computers running various popular operating systems, namely Windows 8.1, Mac OS X Mavericks 10.9.5, Ubuntu 14.04.1 LTS, iOS 7.1.2, and Android KitKat 4.4.4. Potential artefacts recovered during the research include data relating to the installation and uninstallation of the cloud applications, log-in to and log-out from Symform account using the client application, file synchronisation as well as their timestamp information. This research contributes to an in-depth understanding of the types of terrestrial artefacts that are likely to remain after the use of cooperative storage cloud on client devices.

KEYWORDS: forensic science, digital forensics, mobile app forensics, mobile device forensics, cloud forensics, peer-to-peer forensics, cooperative cloud, Symform analysis

Cloud computing is, arguably, one of the most discussed computing paradigms in recent years, due to its popularity among individual consumers and organisations. Gartner (1) forecasted that the cloud computing market will hit US$250 billion by 2017 as cloud adoption increases in organisations. The International Data Corporation (IDC) (2) also published a similar forecast, which indicated that the worth of the cloud computing market will exceed US$107 billion and drive 17% of the IT product spending by 2017. In another survey, Cisco (3) foresees annual global cloud IP traffic to reach 6.5 ZB (541 EB per month) by the end of 2018, up from 1.6 ZB per year (137 EB per month) from 2013.

Despite the promising economical and technological opportunities, cloud storage services are being exploited by criminals, both traditional and cyber ones (4), in several ways such as information theft (5-8) or distributing copyright or illegal materials. Cloud servers have also been exploited as an avenue for conducting denial of service attacks (9, 10), cracking passwords (11), hiding criminal tracks (7) and other criminal endeavours.

Forensic investigations in the cloud environments can be challenging as the data can be distributed across multiple data centers spanning multiple jurisdictional areas, which could potentially inhibit the transfer of evidential data due to the lack of cross-nation legislative mechanisms (12-19). Even if the source of evidence could be identified, it could be illegal to access the raw log data that contains records of multiple users in a multi-tenancy cloud environment located in an overseas jurisdiction (20). The wide range of mobile devices (21) and the use of encryption by CSPs or individuals (13) further complicate cloud forensic investigations.

To ensure the most effective collection of the cloud computing artefacts, it is imperative that forensic practitioners are cognisant about different types of cloud products (or have access to

such information), as well as the potential artefacts detectable on each platform. Depending on the cloud storage solution in use, evidence (i.e., logs) of cloud usage could be recovered from the client devices (22-30). Hence, we seek to identify potential terrestrial artefacts that may remain after the use of Symform cooperative storage cloud (31).

Contribution

Similar to the approaches of Quick and Choo (24-26), we attempt to answer the following questions in this research:

1. Does the act of file download or file upload using Symform cooperative storage cloud alter the file contents and timestamps of the original files?

2. What artefacts can be found on a computer hard drive and memory after a user has used the Symform client application and web application? where are their locations on Windows 8.1, Ubuntu 14.04 LTS, and Mac OS X Mavericks 10.9.5?

3. What data remains on an Apple iPhone 4 and an HTC One X after a user has used the Symform client apps? where are their locations on iOS Version 7.1.2, and Android KitKat 4.4?

4. What data can be seen in network traffic?

Findings from this research will contribute to the forensic community's understanding of the types of terrestrial artefacts that are likely to remain after the use of cooperative storage cloud on devices (i.e., personal computers and mobile devices) running different operating systems.

Organization

The structure of this paper is as follows: In the next section, we outline the background of cooperative cloud storage and related work on cloud forensics. The third section discusses the research methodology and experiment environment and setup. In the "Collection and Timestamp Analysis" section, we detail the evidence collection phase, which will answer the first research

question. Then, we discuss findings from the technical experiments in the "Symform Analysis on Desktop Clients" and "Symform Analysis on Mobile Devices" sections, respectively. These sections will answer the second and third research questions. In the "Network Analysis" section, we discuss analysis of the network traffic, which will answer the final research question of this research study. Finally, we conclude the paper and outline potential future research areas.

Background

A cooperative storage cloud is a peer-to-peer (P2P) cloud storage service model that delivers cloud storage by aggregating the storage space from client end nodes (gaming consoles, laptop, personal computers etc.), eliminating the need for a storage server (32). The model was first implemented by Symbiotic Storage Platform (Symform) in 2009 to provide a cost effective, reliable, and secure cloud storage system (31). The Symform model requires the user to contribute the unused local storage space to the service storage and, in exchange, receiving an amount of bonus storage (in addition to the initial 10GB free storage space) based on a 2:1 ratio of their contributed storage space (33).

The Symform technology works by firstly encrypting each folder with 256-bit Advanced Encryption Standard (AES) encryption locally, and thus only clients in possession of the unique folder encrypted key can access the sync folder. Then, Symform breaks the encrypted files into a series of 64MB blocks, depending on the file size. The process is followed by shredding each block into 64 1MB fragments, before adding 32 redundant storage parity fragments to the original fragments using the patented RAID-96 technology (34). This improves reliability by allowing the system to rebuild the data from the redundant parity fragments in cases when a contribution node holding the original fragments is down. Finally, Symform distributes all the 96 fragments (for each block) in parallel across 96 contribution nodes comprising random devices

located in 150 countries (34), increasing the storage security. The fragments are stored in *%root%\SymformContribution* on a Windows client or */SymformContribution/* on a NAS, Mac, or Linux client by default (35). The resource distribution is monitored continuously by the proprietary Cloud Control backend service, which is hosted by the Amazon Web Services (AWS) (34).

Similar to other cloud storage services, Symform service can be accessed using a web browser (but limited to the downloading, viewing and deleting the files) or a client application available for devices running Microsoft Windows, Apple Mac OS X, Linux, Apple iOS, Android, and Blackberry. Unlike most cloud storage services, Symform allows users to selectively backup any folder across different devices. Symform users are required to install the client application to setup and enable file synchronisation. The user interface for the Linux and Mac (in alternative) client application is a web-based Graphical User Interface (GUI) known as the Remote Device Manager (RDM), which is accessible through *localhost:59234* by default. The default Symform backup folders created by the OS are Music, Pictures, Desktop, and Documents folders; which can be modified by the users. Symform, however, does not provide the ability for file sharing.

Related Work

Since the early 2010's, a number of scholars have highlighted operational and legal challenges and various research opportunities associated with cloud forensic investigations (36-45). In recent years, a number of researchers have published a number of technical solutions to mitigate the identified challenges, particularly those associated with the remote collection of data artefacts from a decentralised cloud infrastructure (42, 46-48). There have also been studies exploring the potential of collecting evidence from client devices (22-30). Other research efforts include:

- Evaluation of the effectiveness of commercial forensic tools (i.e., Guidance EnCase, the Forensics Tool Kit (FTK), Memoryze, and AWS Export) in acquiring evidence remotely from the Amazon EC2 servers (49, 50).
- Determining whether the act of downloading data from the client and web applications of popular cloud services (i.e., Dropbox, Google Drive, and Microsoft SkyDrive) affect the integrity of the data collection process, such as change in the MD5, SHA1, and timestamp information (17).
- Proposal of frameworks, guidelines and methodologies with the aim of providing a systematic approach for forensic collection of cloud artefacts from servers and/or client devices. Martini and Choo (51) were the first to propose a cloud forensic framework, which was derived based upon the frameworks of McKemmish (52) and NIST (53). The framework was used to investigate ownCloud (54), VMWare (23), and XtreemFS (55). Subsequently, Quick and Choo (24-26) and Quick et al. (18) extended the four-stage framework and validated using SkyDrive, Dropbox, Google Drive and ownCloud. Chung et al. (12) proposed a cloud investigation guideline and utilised it to investigate Amazon S3, Google Docs, and Evernote on Windows, Mac OS, iOS, and Android devices. Farina et al. (27) investigated the artefacts left by Bit Torrent Sync and outlined an investigative framework for the remote collection of evidence from a decentralised file synchronisation network. Scanlon et al. (56) further extended the work of Farina et al. (27) and designed a methodology for the network investigation of Bit Torrent Sync (57). More recently in 2015, Do, Martini and Choo (58) proposed an adversary model for digital forensics, and they demonstrated how

such an adversary model can be used to investigate mobile devices (i.e. Android smartwatch (59)) and apps.

- Proposal of a conceptual forensic-by-design framework (60) designed to integrate forensics tools and best practices in the development of cloud systems.

Due to the recency of cooperative storage cloud services, this is the first forensic research undertaken to identify artefacts of forensic interest that may remain after the use of such services on the client's device.

Research Methodology

In this section, we provide an overview of the cloud investigation framework used to guide the investigations in this paper as well as the experimental setup.

Cloud Investigation Framework

It is essential that (digital) forensic investigators or practitioners adhere to generally accepted forensic principles, standards, guidelines, procedures and best practices when undertaking digital forensic investigations (16, 61). In particular, Kent et al. (53) (p.5) define the forensic process as follows:

"An individual performing forensic activities needs to understand forensic principles and practices, and follow the correct procedures for each activity, regardless of which group he or she is a member."

As an example, Mckemmish (52) explained that digital forensic investigations should be based on four principles, namely minimal of the original, account for any changes, comply with the rules of evidence, and not to exceed knowledge. Similarly, the digital forensic principles of the United Kingdom Association of Chief Police Officers (ACPO) specified that: no action should change data, when it is necessary to access original data the persons accessing data should

be competent to do so, a record of processes should be made, and the investigator in charge is responsible to ensure the principles are adhered to (61). Moreover, the National Institute of Standards and Technology (NIST) prescribed that a digital forensics framework should contain the necessary components, namely collection, examination, analysis, and reporting (53).

In this research, we adopt the cloud investigative framework proposed by Martini and Choo (51) (see Figure 1). While the framework shares several similarities with the frameworks of McKemmish (52) and NIST (53), it differs in a number of ways. The primary difference being that of the third phase, which emphasises one or more simultaneous iteration(s) of the framework with evidence source identification and preservation via the associated nodes. In the following, we briefly explain each of the four investigation phases in the context of our research.

1. *Evidence source identification and preservation*: In the first phase, we identified the physical hardware of interest, which contained the virtual disk (VMDK) and memory files (VMEM) in each VM folder. The mobile devices used in this research were HTC One X running Android KitKat 4.4.4 and Apple iPhone 4 running iOS Version 7.1.2. We then created a forensic copy of the VMDK and VMEM files in the E01 container and raw image file (dd) formats respectively. For the mobile devices, we acquired a bit-for-bit image of the internal storage and converted the images to the E01 container format. For all the forensic images created, an MD5 and SHA1 hash value was calculated and subsequently verified with the original copies.

2. *Collection*: In this phase, we collected files containing the details needed for analysis and keyword searching in the forensic copies. Similar to the earlier evidence source identification and preservation phase, we calculated the MD5 and SHA1 hash values of each original file and subsequently verified each collected or exported file. Further details of this phase are explained in the “Symform Analysis on Desktop Clients” section.

3. *Examination and analysis*: This phase is concerned with the assessment and extraction of the evidential information from the collected data. The analysis included identifying the sync, file management, and authentication metadata, cloud transaction records, data storage, as well as the relevant timestamp information useful for establishing connections between the suspect and the crime.

4. *Reporting and presentation*: This final phase relates to the preparation and presentation of the information resulting from the analysis phase. A summary of the evidence findings is outlined in the "Concluding Remarks" section.

Experimental Setup

For our experiments, we created a total of 33 virtual machine (VM) snapshots each one representing different physical systems to simulate a series of real life scenarios of using Symform (i.e., install, access, upload, download, view, delete, and uninstall) on various operating systems (OS), as detailed in Table 1. The VMs were hosted using VMware Fusion Professional version 7.0.0 (2103067) on a Macbook Pro (Late 2012) running Mac OS X Mavericks 10.9.5, with a 2.6GHz Intel Core i7 processor and 16GB of RAM. As explained by Quick and Choo (24-26), using physical hardware to undertake setup, erasing, copying, and re-installing would have been an onerous exercise. Moreover, a virtual machine allows room for error by enabling the test environment to be reverted to a restore point if the results are unfavorable. The workstations were configured with minimal space in order to reduce the time required to analyse the considerable amounts of snapshots in the latter stage. Immediately upon the completion of each experiment, we took a snapshot of each VM prior to and after being shutdown in order to allow restoring at a later stage, if necessary. This also allowed the capture of the volatile memory .VMEM files at a later stage in the former. The decision to instantiate the

hard disks and physical memory dumps with the virtual disk and memory files was to prevent the datasets from being adulterated with the use of memory/image acquisition tools (24-26).

With regard to the mobile app experiments, we used a default factory restored (physical) iPhone 4 running iOS 7.1.2 and an HTC One X running Android KitKat 4.4.4. The decision was guided by the consideration that running a mobile emulator on a computer desktop may omit the hardware features of a physical mobile device, leading to unrealistic information (62). The mobile devices were jailbroken/rooted with 'Pangu8 Version 1.1' and 'Odin3 Version 185' (respectively) to enable root access to the devices. To this regard, iOS 8 was not used due to the unavailability of the jailbreak tool at the time of this research. A binary image was made of the mobile devices for different Symform usage scenarios using 'dd' over SSH/ADB Shell. In particular, we took the first image prior to the installation of the Symform apps for the base image, ensuring that neither the Symform nor the Enron data were on the devices. Then, we installed the Symform iOS app Version 1.13 and Android app version 1.3 on the respective devices and took the second image of the devices. The third image was taken after viewing the dataset files (the only feature supported by the mobile apps) in the Symform apps. Finally, we took the last image following the uninstallation of the apps.

Similar to the approaches of Quick and Choo (24-26), the 3111th and 13100th email messages of the UC Berkeley Enron email dataset (downloaded from http://bailando.sims.berkeley.edu/enron_email.html on 24th of September 2014) were used to create the sample files and saved in .RTF, .TXT, .DOCX, .JPG (print screen), .ZIP, and .PDF formats, providing a basis for replicating the experiments in future. Wireshark was deployed on the host machine to capture the network traffic from the client workstations/devices for each

scenario. The experiments were repeated thrice (at different dates) to ensure consistency of findings. Table 2 details the tools prepared for the evidential analysis.

Collection and Timestamp Analysis

Before undertaking the evidential analysis, we collected test data that matched the search terms 'symform' and 'enron' in the hard disk images, but held formats unsupported by the Autopsy forensic browser for analysis using the tools of relevance in the latter phase. These included SQLite database files, PLIST files, prefetch files, event logs, shortcuts, thumb- nail cache, $MFT, $LogFile, $UsnJrnl, as well as web browsers' data folders/files (i.e., *%AppData%\Local\Google*, *%AppData%\Local\Microsoft\Windows\WebCache*, *%AppData%\Roaming\Mozilla*, *%AppData%\Local\Microsoft\Windows\Temporary Files\index.dat*). The volatile data was collected using the Volatility tools, Photorec file carver, and HxD Hex Editor for the physical memory dumps, and Wireshark and Netminer network analysis software for the network captures.

Whilst undertaking keyword search for the data of relevance, we determined that there was no data related to Symform and the Enron emails on the control base VM snapshots (1.0, 1.1 IE, 1.2 MF, 1.3 GC, 2.0, and 3.0). This suggested that the Symform/Enron related data located in the remaining snapshots were remnants from Symform use. An inspection of the metadata of the downloaded files on the Windows 8.1 client observed that the last accessed and modified timestamps were the times when the files were downloaded, and the last written timestamps retained its original value unchanged. On the Ubuntu client, the added timestamps were the times when the files were downloaded, while all other timestamps (i.e., modification, creation, and last opened) remained unchanged. As for the Mac OS client, only the accessed timestamps matched the file download times; the modification timestamps preserved its original timestamps. In all

cases, we determined that the MD5 and SHA1 hash values for the downloaded files were similar to the that of the original copies, suggesting that no alteration was made during the file transfers.

Symform Analysis on Desktop Clients

In this section, we present the findings of our Symform analysis on Windows 8.1, Ubuntu 14.04.1 LTS, and Mac OS X Mavericks 10.9.5.

Directory Listings and Files of Forensic Interest

Analysis of the directory listings determined that the install directory is in *%Program Files%\Symform*, */opt/symform/*, and */Library/Application Support/Symform/* on the Windows 8.1, Ubuntu, and Mac OS clients (respectively) by default. Of all the desktop clients investigated, records of the sync, file management, and authentication metadata were predominantly located in the node.config XML file in *%Program File%\Symform\Node Service* on the Windows 8.1 client, */SymformContribution/* and */var/lib/symform/* on the Ubuntu client, and */private/var/lib/symform/* on the Mac OS client. The records comprised the server's URL (prefixed with 'serverAddress'), unique SHA-1 node ID for the local machine (prefixed with 'nodeId' in the 'node' property), encrypted secret key (prefixed with 'secretKey=' in the 'node' property) in base64 format, login username or email address (prefixed with 'username=' in the 'userCredentials' property), and encrypted password (prefixed with 'password=' in the 'userCredentials' property) in base64 format. The node.config file also held the folder IDs (prefixed with 'remoteFolderGlobalId'), folder paths (prefixed with 'localPath='), remote folder names (prefixed with 'remoteFolderName='), and read-write permission information (prefixed with 'direction=') for the sync folders added to the local machine; each folder created an opening and closing folder sub-tag in the 'folderMapping' property. The remote folder name is the root folder name where the sync folder is connected to. In all cases, initialising a Symform sync

folder created two hidden sub-directories (*.symform* and *.symform-store*) in the sync folder. The file of particular interest being the *%.symform%/metadata* sqlite3 database, which cached the filenames, sizes, last modified times, and checksums for the synced files in the FolderItem table; Figure 2 shows an example.

Examination of the Windows 8.1 client determined that the files downloaded through the web browsers were stored in *%Downloads%* by default. Each downloaded file was given an Alternate Data Stream (ADS) ZoneTransfer marker (ZoneID) with reading 'ZoneID=3', indicating that the files were downloaded from Internet zone (63). The ADS ZoneID can prove useful to determine the origin of a synced file especially in the absence of the Internet browsing history.

On the Mac OS client, additional cloud transaction records could be recovered from the */Users/<User Profile>/Library/Caches/com.symform.mac.Symform/Cache.db*. Similarly to the structure of the Safari browser's Cache.db database (64), the cached items were stored in the 'receiver_data' table column of the cfurl_cache_receiver_data table, and the corresponding URLs and timestamps could be located in the cfurl_cache_response table, in the 'request_key' and 'time_stamp' table columns respectively. By issuing the SQL query "*SELECT cfurl_cache_receiver_data.receiver_data, cfurl_cache_response.request_key, cfurl_cache_response.time_stamp FROM cfurl_cache_receiver_data, cfurl_cache_response WHERE cfurl_cache_receiver_data.entry_ID=cfurl_cache_response.entry_ID*" (64), it was possible to recover caches of the HTTP requests/responses for the node configuration property as well as sync and Google Analytic services from the Cache.db database, alongside the URLs and the associated timestamps. Figure 3 illustrates that running and stopping of the Symform service produced the URLs *http://localhost:<PortNumber>/syncService/start* and

http://localhost:<PortNumber>/syncService/stop respectively, and the timestamps of which could indicate the last run time.

Further analysis of the Cache.db database determined that the HTTP request for the node configuration property produced the URL *http://localhost:<PortNumber>/nodeconfig*, and we could recover the complete node.config content from the receiver data table column of the cfurl_cache_receiver_data table, providing potential for alternative methods for recovering the node.config file. We could also recover the HTTP response body for the metric information request for the sync session, from the cache records that referenced the URL *http://localhost:<PortNumber>/metric*. The property information of forensic interest with the metric information request included number of syncing folders and files, total file size, last sync time, as well as the remote folder name(s), local folder name(s), number of transferred files, and service start times; an example is as follows: *{"NumberOfSyncingFolders":0,"NumberOfSyncingFiles":0,"TotalSizeOfSyncingFiles":0,"Total NumberOfSyncingFiles":0,"TransferSpeed":0.0,"TransfersRunningCount":0,"LastHeartbeat":"2014-11-16T00:57:52.656653Z","FolderMetrics":[{"RemoteFolderName":"Desktop","LocalFolderName":"/Users/alice/Desktop","NumberOfFilesPending":0,"CurrentFileSyncBatchSize":6,"CurrentFileSyncBatchCompleted":6,"SizeOfCommittedFiles":0,"NumberOfCommittedFiles":0,"Errors":null,"FilesInProgress":[],"SyncStatus":1,"LastReported":"2014-11-16T00:57:52.656653Z"}],"DailyRateOfChange":0,"EstimatedSyncTime":"00:00:00","ServiceStartTime":"2014-11-16T00:57:35.661268Z"}*.

Of all the desktop clients investigated, deleting the synced files did not remove the caches in the *%.symform%/metadata* sqlite3 database. Records of the deleted files could be differentiated from the ‘Size’ and ‘Checksum’ table columns given the value *nil*. Further analysis

determined that copies of the deleted files could be recovered from the non-emptied Recycle Bin/Trash directory, along with the original file extensions in *%$Recycle.Bin%\SID*, */home/<User Profile>/.local/share/Trash/files*, and */Users/<User profile>/.Trash* on the Windows 8.1, Ubuntu, and Mac OS desktop clients correspondingly. In the circumstances when the files were downloaded using a web browser, it was also possible to recover the ADS ZoneIDs from the Recycle Bin directory of the Windows client. However, it is noteworthy that the deleted files will be renamed to $R followed by a set of random characters in the Recycle Bin directory, and hence a manual matching of the file metadata (i.e., original paths, sizes, and delete times) from the $I files may be required to determine the sync files. On the Ubuntu client, the original path and deletion time information could be located for the deleted files in the .TRASHINFO files in */home/<User Profile>/.local/share/Trash/info/* – see Figure 4. We also managed to recover the deleted files from the unallocated space intact.

Undertaking uninstallation of the Symform Windows application revealed that the install and data directories *%Program File%\Symform*, *%AppData%\Local\Temp\Symform*, and *%SymformContribution%* remained on the hard drive, but they were empty. Similarly, when the uninstallation occurred on the Ubuntu client, we observed that the directories */opt/symform*, */var/log/symform*, and */var/lib/symform* were emptied. As for the Mac OS client, the uninstallation emptied the directories */Library/ApplicationSupport/symform* and */SymformContribution*. In all cases, it was determined the uninstallation did not remove the *.symform* and *.symform-store* sub-directories from the sync folder, suggesting that there will be references remaining in the directory listing after uninstallation of the client applications to indicate the sync directories.

Log Files

Logs play a vital role in an incident investigation (65-67). By default, the Symform logs were stored in *%Program Files%\Symform\Node Service\logs*, */opt/symform/bin/logs/* or */var/log/symform/*, and */Library/Application Support/Symform/bin/logs/* and */private/var/logs/Symform/* on the Windows, Ubuntu, and Mac OS desktop clients (respectively) unencrypted. The logs were archived hourly in compressed .GZ format, with the exception of the setup logs (i.e., symformsetup.log, symformupdater.log, and loguploader.log). Analysis of the symformsetup.log determined that records of the client application setup and sync folder initialisation could be recovered. The records provided information such as the version of the Symform client application installed, email addresses used to login the client application, login times, as well as the full paths, initialisation timestamps, folder IDs, and folder owners' node IDs associated with the sync folders (see Table 3). In the symformsync.log and symformsync-mono.log, we recovered details of the cloud transactions such as the node configuration property as well as the filename and timestamp information for the uploaded and downloaded files. Table 4 shows the entries of forensic interest from symformsync.log. Alternatively, the node configuration property could be located in the symformcontrib.log and symformcontrib-mono.log, by searching for the term "INFO ContributionHost - <node version=". The timestamp information noted along the log entries could be used for timeline or super timeline analysis (68).

A search for the term symform indicated the installation and uninstallation timestamps in the Application.evtx and System.evtx event logs (located in *%Windows%\system32\config*\) of the Windows 8.1 client; Figure 5 shows an example of the install log from System.evtx. Running the client application logged the service creation event to the System.evtx log, and the event name of which could be differentiated from 'symformsync'. When the Symform installation

occurred on the Ubuntu client, we were able to determine the install time from the Syslog entry "*Nov 16 08:39:59 ubuntu AptDaemon.Worker: INFO: Installing local package file: /home/suspectpc/Desktop/Symform.deb*" at */var/log/syslog*. Alternatively, the install time could be located in the dpkg log */var/log/dpkg.log*, alongside the Symform version information from the entry "*2014-11-16 08:40:29 status installed symform:amd64 4.24.0-1*". The similar install information could be located for the Mac OS client application in the Install log */private/var/log/install.log*, from the entry "*Nov 15 16:55:17 Alices-Mac.local Installer[494]: Symform 4.20.0.0*".

Thumbnail Cache

Thumbnail cache is a potential source of synced images (69). Looking through the Windows thumbcache directory *%AppData%\Local\Microsoft\Windows\Explorer* we located copies of thumbnail images for the client or web application (i.e., Symform logo and image icons appeared on the GUI), indicative of recent Symform usage. When the sample files were synced, it was possible to recover copies of the thumbnail images for the synced image and PDF files from the Windows thumbcache directory. However, the installation and file synchronisation did not produce a thumbnail cache in the Ubuntu and Mac OS clients in our experiments.

Web Browser Artefacts

Web browsing record is another potential source of information in cloud investigations (24-26, 54). Whilst accessing the web application, we observed that the username could be located at the top right corner of the browser. The web application retained a list of devices/nodes associated with the account in the left-hand pane of the browser. When hovered over an inactive device (marked with 'X'), we observed the message "Device has not been reported in X days". Unlike Dropbox and Google Drive cloud services studied in (24-26), other

than the duration (in days) from the last modified date, the Symform web application does not show the last accessed, creation, and written times associated with the backup files. The files deleted in the past 15 days could be restored using the "show deleted" option in the sync folder(s).

Examination of the web browsing history (of all the web browsers investigated) determined that the login and download URLs could be differentiated from *control.symform.com* and *content.symform.com/api/v0/folder/<Folder ID>/<Filename>* respectively. Setting up the Ubuntu client application using the RDM produced the URLs *127.0.0.1:59234/tour*, *127.0.0.1:59234/setup*, and *127.0.0.1:59234/setup/done* in the browsing history. When the login occurred in the RDM, we observed the URLs *127.0.0.1:59234/login* and *127.0.0.1:59234/general*. In all cases, the web browsing history also held the associated visit timestamp and view count information.

Analysis of the web caches (of all the web browsers investigated) determined that the downloaded files could be recovered. The web caches also held the images and HTML documents for the Symform web application, and the timestamps of which could reflect the access times. The login credentials (if manually saved) could be recovered using Nirsoft Web Browser Passview (70).

Memory Analysis

Memory forensics enables practitioners to recover data that might otherwise be lost if a device is powered down (71, 72). The memory analysis in this research encompassed data carving using Photorec, keyword searching using Hex Workshop, and contextualising the RAM contents using Volatility. Examinations of the running processes using the 'pslist', 'linux_pslist', and 'mac_pslist' functions of Volatility revealed the process names, process identifiers (PIDs),

and parent process identifiers (PPIDs) for the Symform services, which included the process initiation times. The process names of which could be discerned from 'symformstatus.exe', 'symformupdater.exe', 'symformcontrib.exe', and 'symformsync.exe' on the Windows 8.1 client; 'symformstatus', 'symformupdater', 'symformcontrib', and 'symformsync' on the Ubuntu client; only 'symform' on the Mac OS client; Figure 6 shows an example of the 'pslist' output for the Windows client application. Analysing the network details using the 'netscan', 'netstat_linux', and 'netstat_mac' functions of Volatility, we recovered the network information associated with the processes such as the IP addresses of the local, foreign, and peer nodes, alongside the port numbers and socket states, providing potential for alternative methods for recovering the network information.

A manual analysis of the memory dumps of the file synchronisation VM snapshots (1.4.2, 1.4.3, 2.3, 2.4, 3.3, and 3.4) revealed copies of the files of forensic interest (i.e., symformsync.log, node.config, synced files, as well as metadata database) in the memory space of 'symformsync.exe' on Windows machine, 'symformsync' on Ubuntu machine, and 'symform' on Mac OS machine in plain text; useful when seeking to determine the origin of the texts in the absence of the original files. Figure 7 shows an example of the remnants from metadata database. A search for the entries unique to the files could enable future searches of the remnants of relevance. When the downloads occurred in a web browser, we could recover copies of of the login and file download URLs from the memory dump, by searching for the URLs such as *control.symform.com* and *content.symform.com*. The URLs appeared to be remnants from the web browsing history. When the synced files were deleted using the web application, we could recover copy of the file deletion message " Are you sure you want to delete <Filename> from your account? Files will be saved for up to seven days for recovery?" from the memory dump,

indicating the filenames for the sync files recently deleted from the desktop clients. However, there was no timestamp information to inform the deletion time.

When the logins occurred on a web browser, we could recover the username and password from the memory dump in plain text, prefixed with the terms ‘session.emails’ and ‘session.password’ respectively. As for the RDM, it was observed that the login username and password could be located following the terms ‘email": and ‘password”: respectively. We speculate that the credential details were remnants from the login payloads. When the credentials were saved in the Mac OS client’s Keychain credential manager, it was possible to recover the master key using the keychaindump function of Volatility. The master key could then be used to decrypt the credential details using chainbreaker.py script, as shown in Figure 8.

Data carving of the memory dumps (using the default settings) determined that only the Enron dataset files as well as image icons, HTML documents, and script files used by the client/web applications could be recovered, since the files maintained a general header and footer file structure. However, we identified that the node.config and metadata files could be manually carved using the header and footer information of “*3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31 2E 30 22 20 65 6E 63 6F 64 69 6E 67 3D 22 75 74 66 2D 38 22 3F 3E...3C 2F 6E 6F 64 65 3E 3C 2F 63 6F 6E 66 69 67 75 72 61 74 69 6F 6E 3E*” and “*53 51 4C 69 74 65 20 66 6F 72 6D 61 74 20 33...69 6E 64 65 78 73 71 6C 69 74 65 5F 61 75 74 6F 69 6E 64 65 78 5F 46 6F 6C 64 65 72 49 74 65 6D 5F 33 46 6F 6C 64 65 72 49 74 65 6D 10*” (respectively).

Windows Registry

The registry provides a rich source of information about a Windows program (73). Although five hives could be seen in the registry, only HKEY USERS(HKU) and HKEY LOCAL MACHINE (HKLM) hives are tangibly real, since the remaining hives are merely

symbolic links to the two master keys (74). Examination of the Windows client's registry revealed the Symform version installed, install path, install date, and other relevant information in the *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\<Product GUID>\InstallProperties* and *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\<Product GUID>* registry branches. However, both the registry branches were removed after uninstallation of the client application.

Similar to any other Windows application, analysis of the *Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32* registry revealed the last accessed time of the loader file in the 'CIDSizeMRU', 'OpenSavePidlMRU', and 'LastVisitedPidlMRU' registry subkeys, where MRU is the abbreviation for Most-Recently-Used. The findings suggested that the client application was recently used, had been opened or saved within a Windows shell dialog box, and was used to open the files documented in the 'OpenSaveMRU' subkey, respectively (75).

An inspection of the *Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs* registry indicated the full path references to the loader, HTML document for the web application (Symform Web App.htm), Enron dataset files, along with the last accessed times, suggesting that the files were recently executed or opened through Windows Explorer (76). Typing the Symform URLs in the Internet Explorer web browser produced references to the URLs alongside the access times in the *Software\Microsoft\Internet Explorer\TypedURLs* and *Software\Microsoft\Internet Explorer\TypedURLsTime* registry branches (respectively). It is to the best of the authors' knowledge that none of the remaining browsers utilise the registry in the way that Internet Explorer does.

Prefetch Files

A prefetch file contains information about a loaded Windows application, such as the filename and full path for the executable file, number of times the application has been loaded, associated dynamic-link library (DLL) files, last run time, and other asso- ciated timestamps. Examination of the prefetch files in *%SystemRoot%\Prefetch* determined that the installation created the prefetch files SYMFORMSETUP.EXE.pf, SYMFORMNODENEW.EXE.pf, and SYMFORMUPDATER.EXE.pf. When file synchronisation occurred, we observed the additional prefetch files such as SYMFORMCONTRIB.EXE.pf (for the contribution service), SYMFORMSTATUS.EXE.pf, and SYMFORMSYNC.EXE.pf, but no prefetch entries were located for the Enron dataset files. The Symform prefetch files remained even after uninstalling the client application.

Link Files

Link (.lnk) files are shortcut metadata files used by Windows to maintain a list of linked paths, timestamps, and file sizes associated with a file accessed on the file system (77). The information can be useful when seeking to determine the origin (original path) of a file in the circumstances when the file is moved or deleted, including the creation, modified, and last accessed timestamps. In our research, it was determined that the Symform installation created two link files Symform Setting.lnk and Symform Status.lnk for *%Program Files%\Symform\Node Service\symformsetup.exe* and *%Program Files%\Symform\Node Service\symformstatus.exe* (respectively) in *%ProgramData%\Microsoft\Windows\Start Menu\Programs\Symform*. Analysis of the file synchronisation, delete, and uninstall VM snapshots (1.4.2, 1.4.2.1, 1.4.3, and 1.4.3.1) located link files for the Enron dataset files in the

Recent Documents directory (*%AppData%\Roaming \Microsoft\Windows\Recent*), indicating that the files were recently accessed on the file system.

Symform Analysis on Mobile Devices

The data directory of the Symform iOS app could be located at */private/var/mobile/Applications/<Universally Unique Identifier (UUID) for the Symform iOS app>*. An inspection of the iTunesMetadata.plist file in the data directory indicated the purchase date from the purchaseDate property. Within the */private/var/mobile/Applications/<UUID for the Symform iOS app>/Library/Caches/com.symform.ios.Symform/Cache.db* there held caches of the HTTP requests/responses associated with the cloud transactions, similarly to those located for the Mac OS client.

Analysis of the Android client determined that data directory could be located at */data/data/com.symform.android.symform*. The file of particular interest with the data directory was the */data/data/com.symform.android.symform/shared prefs/SymformPrefs.xml* file, which held the login credentials (i.e., email address and encrypted password) associated with the app. The credential information can assist a practitioner to ascertain whether a user has logged in from the app.

For both the mobile apps, viewing the synced files produced copies of the viewed files in */private/var/mobile/Applications/<UUID for the Symform iOS app>/tmp/downloads/* and */data/data/com.symform.android.symform/files/downloads/* of the iOS and Android clients (respectively), which included the original file extensions and last view timestamps. However, the files were renamed to a set of random characters in the iOS client. Uninstallation of the mobile apps removed the data directory completely from the mobile clients.

Network Analysis

Accessing the login webpage produced the IP address *173.193.191.132* (in our research) over port 80 (HTTP), with the URL referencing control.symform.com. As soon as the logins took place, we observed that the connections were established on port 443 (HTTPS), and the certificates were provided by Starfield Technologies (78). The next IP addresses accessed were *54.231.*.**) (registered to Amazon Technologies, Inc), which we theorised for accessing the Cloud Control backend service hosted by the AWS.

Undertaking file download using the web application, we could estimate the download times from the timestamps of the TCP packets that referenced the URL *content.symform.com*. When the file synchronisation occurred on the desktop clients, we identified UDP as the carrying protocol. A closer inspection of the packet details revealed the IP addresses of the peer nodes, but the port numbers appeared to be random, thereby making the ports unpredictable. Although there is currently no method known outside the client application to rebuild the synced files from the encrypted file fragments/encrypted traffic, we could locate remnants of the HTTP requests for the file fragments in the UDP stream (see Figure 9). The file fragments were represented by a unique ID in the form of <Unique SHA-1 for a file fragment>.<File fragment number>.<Folder ID>. The finding suggested that a practitioner can match the folder IDs (determined from the files of forensic interest such as node.config, symformsetup.log, and symformsync.log) with the file fragment IDs to determine the sync time(s) associated with a sync folder. Rebuilding of the network captures only recovered the HTML documents, script files, and image files from the unencrypted traffic.

Concluding Remarks

In this paper, we described the terrestrial artefacts from the use of Symform cooperative storage cloud service on a Window 8.1, Ubuntu 14.04.1 LTS, Mac OS X Mavericks 10.9.5, iOS 7.1.2 and Android KitKat 4.4.4 desktop/mobile client. In our case study, we determined that a practitioner can commence the desktop client forensics by analysing the node.config file for the sync, file management, and authentication metadata. These include the login email, which could inform a practitioner whether a user had logged in Symform from the target system; directory paths, folder names, and IDs for the sync folders, which could enable a practitioner to identify the sync directories as well as correlate the metadata with the cloud transaction records (in the metadata database and Symform logs) to determine the filenames for the synced files and the corresponding sync times; node ID for the local device, which could be used to match the Symform logs to identify the locally created sync folders. Additionally, the node and folder IDs could assist a practitioner in correlating any external data that might have been obtained from an Internet service provider (ISP) or other external content or service provider.

Similar to any other client applications, our examinations of the OS-specific logs, thumbnail cache, and Windows' shortcuts, $LogFile, $MFT, $UsnJrnl, as well as registry (i.e., recentdocs, run, and userassist) revealed that additional timestamp information could be recovered to support evidence found in all scenarios. The files that were deleted could be potentially recovered from the non-emptied Recycle Bin/Trash directory and unallocated space, but the results may not be definitive. Our examinations of the mobile clients determined that the viewed files could be recovered, which could assist a practitioner to ascertain whether a user has accessed a file using the mobile app. However, only the iOS client cached the cloud transaction records.

Our analysis of the web browser artefacts determined that a practitioner can identify the login and file download times, including the view count information by searching for the URLs of relevance in the web browsing history. The download URLs could be used to match the web browsing caches for copies of the downloaded files. When the login credentials were saved in the web browsers, we determined that it is possible to recover the login email address and password using a password recovery tool for web browsers. Our findings also suggested that when a user has accessed Symform using a web browser, there will be URL references remaining in *the /%Default%/Current Tabs* and */%Default %/Last Session* files of Google Chrome borwser, */%PROFILE%.default/sessionstore.js* of Mozilla Firefox browser, as well as *%AppData%\Local\ Microsoft\Windows\WebCache\V01.log* and *%AppData%\Local\Microsoft\Windows\WebCache\ WebCacheV01.dat* of Internet Explorer browser to indicate recent Symform usage. In all cases, we determined that the MD5 and SHA1 hash values did not change during the process of uploading, storage, and downloading files, establishing the integrity of files downloaded from the client/web applications.

Although data from the application layer were encrypted in the network traffic, when the file synchronisation took place (via the client application), it was possible to determine from the file fragment IDs the sync times relating to the sync folders in the UDP traffic. Hence, we recommend that the network captures be undertaken wherever practical. Our examinations of the physical memory captures indicated that the memory dump can provide potential for alternative methods for recovering the login credentials, node.config file, Symform logs, and metadata database in plain text. A search for the terms or header/footer structure unique to the files of relevance as identified in our research could enable future searches. The memory dump could also provide an alternative method for recovering the running process and network information

using the 'pslist' and 'netscan'/'netstat' functions of Volatility (respectively). The PIDs could assist the investigator in obtaining data associated with the Symform client application during further analysis of the physical memory dumps (i.e., locating the data remnants associated with the process using the Yarascan function of Volatility). The presence of the artefacts in the memory dump also means the artefacts could be potentially located in the swap files as a result of inactive memory pages being swapped out of the memory to the hard disk during the system's normal operation (24-26, 72). Nevertheless, a practitioner must keep in mind that memory changes frequently according to users' activities and will be wiped as soon as the system is shut down. Hence, obtaining a memory snapshot of a compromised system as quickly as possible increases the likelihood of preserving the artefacts before being overwritten in memory.

Collectively, our research suggested that there is currently no method known outside the client application or CSP that enables reconstruction of the synced files from the file fragments. Hence, underlining the importance of the client forensics. Table 5 summarises the key artefacts located in our research. At the time of this research, findings are accurate to the best of the authors' knowledge. However, new releases of forensic tools, hardware (i.e., mobile devices, personal computers and cloud servers) and operating systems (i.e., virtual machine managers) may change the way the availability and recoverability of key artefacts in the future.

To keep pace with technological advances, future work would include extending this research to other popular cooperative storage cloud services (i.e., Storj), as well as developing a forensically sound tool to automate collection of artefacts common to popular cooperative storage cloud services.

References

1. Forecast: it services, 2011-2017, 4q13 update; https://www.gartner.com/doc/2637515/forecast-it-services--q (accessed November 20, 2014).
2. Worldwide and regional public IT cloud services 2013-2017; http://www.idc.com/getdoc.jsp?containerId=242464 (accessed November 20, 2014).
3. Global cloud index: forecast and methodology 2013-2018 white paper; http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html (accessed November 30, 2014).
4. Choo KKR. Organised crime groups in cyberspace: a typology. Trends in Organized Crime 2008 Sep 1;11(3):270–95.
5. Choo KKR. Cloud computing: challenges and future directions. Trends and Issues in Crime and Criminal Justice 2010 Oct;(400):1.
6. 5 things to know about the celebrity nude photo hacking scandal; http://edition.cnn.com/2014/09/02/showbiz/hacked-nude-photos-five-things/ (accessed November 18, 2014).
7. Sony network breach shows amazon clouds appeal for hackers; http://www.bloomberg.com/news/articles/2011-05-15/sonyattack-shows-amazon-s-cloud-service-lures-hackers-at-pennies-an-hour (accessed June 5, 2014).
8. Symantec. The Trojan hydraq incident: analysis of the aurora 0-day exploit; http://www.symantec.com/connect/blogs/trojanhydraq-incident-analysis-aurora-0-day-exploit (accessed November 20, 2014).

9. Cloud-based denial of service attacks looming, researchers say; http://www.darkreading.com/smb-security/167901073/security/perimeter-security/226500300/index.html (accessed November 27, 2014).
10. Osanaiye O, Choo KKR, Dlodlo M. Distributed denial of service (DdoS) resilience in cloud: Review and conceptual cloud DdoS mitigation framework. Journal of Network and Computer Applications 2016;67:147–65.
11. Amazon EC2 helps researcher to crack wi-fi password in 20 minutes; http://www.ibtimes.com/articles/100314/20110112/amazon-ec2-password-wi-hacking-cracking-brute-%20force-attack-%20wpa-psk-encryption-cloud-computing-iaa.htm (accessed November 20, 2014).
12. Chung H, Park J, Lee S, Kang C. Digital forensic investigation of cloud storage services. Digital Investigation 2012 Nov 30;9(2):81–95.
13. Grispos G, Storer T, Glisson WB. Calm before the storm: the challenges of cloud. Emerging digital forensics applications for crime detection, prevention, and security. 2013;4:28–48.
14. Hooper C, Martini B, Choo KK. Cloud computing and its implications for cybercrime investigations in Australia. Computer Law & Security Review 2013 Apr 30;29(2):152–63.
15. Martini B, Choo KK. Cloud forensic technical challenges and solutions: a snapshot. IEEE Cloud Computing 2014 Nov;1(4):20–5.
16. N. I. of Standards and T. (NIST). Nist cloud computing forensic science challenges; http://safegov.org/media/72648/nist-digital-forensics-draft-8006.pdf (accessed October 28, 2014).

17. Quick R, Choo KKR. Forensic collection of cloud storage data: does the act of collection result in changes to the data or its metadata? Digital Investigation 2013;10(3):266–77.

18. Quick D, Martini B, Choo KKR. Forensics cloud storage. Waltham, MA: Syngress/Elsevier, 2014.

19. Taylor M, Haggerty J, Gresty D, Almond P, Berry T. Forensic investigation of social networking applications. Network Security 2014;11:9–16.

20. Hogben G, Dekker M. Procure Secure: a guide to monitoring of security service levels in cloud contracts. European Network and Information Security Agency (ENISA) Report 2012 Apr. Crete, Greece: ENISA, 2012.

21. Tassone C, Martini B, Choo KKR, Slay J. Mobile device forensics: a snapshot. Australian Institute of Criminology 2013;460:1–7.

22. Hale JS. Amazon Cloud Drive forensic analysis. Digital Investigation 2013;10(3):259–65.

23. Martini B, Choo KK. Remote programmatic vCloud forensics: a six-step collection process and a proof of concept. Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TRUSTCOM '14); 2014 Sept 24-26; Beijing, China. Washington, DC: IEEE Computer Society, 2014;935–42.

24. Quick D, Choo KKR. Digital droplets: Microsoft skydrive forensic data remnants. Future Generation Computer Systems 2013;29(6):1378–94.

25. Quick D, Choo KKR. Dropbox analysis: data remnants on user machines. Digital Investigation 2013;10(1):3–18.

26. Quick D, Choo KKR. Google drive: forensic analysis of data remnants. Journal of Network and Computer Applications 2014;40:179–93.

27. Farina J, Scanlon M, Kechadi MT. BitTorrent Sync: first impressions and digital forensic implications. Digital Investigation 2014 May 31;11:S77–86.

28. Martini B, Do Q, Choo KKKR. Mobile cloud forensics: an analysis of seven popular android apps. In: K R, Choo K-KR, editors. Cloud security ecosystem. Waltham, MA: Syngress, an Imprint of Elsevier, 2015.

29. Shariati M, Dehghantanha A, Choo KKR. SugarSync forensic analysis. Australian Journal of Forensic Sciences 2016;48(1):95–117.

30. Shariati M, Dehghantanha A, Martini B, Choo KKR. Ubuntu one investigation: detecting evidences on client machines. In: K R, Choo K-KR, editors. Cloud security ecosystem. Waltham, MA: Syngress, an Imprint of Elsevier, 429–46.

31. Symform announces 11 million$ series b with strong business momentum; http://www.symform.com/news/press-releases/symform-wins-two-awards/ (accessed November 21, 2014).

32. Symform announces worlds first cooperative storage ex-change; http://www.symform.com/news/press-releases/symform-announces-worlds-first-cooperative-storage-exchange/ (accessed July 2, 2014).

33. Affordable cloud storage pricing plans; http://www.symform.com/plans-pricing/ (accessed November 28, 2014).

34. Revolutionary cloud architecture; http://www.symform.com/resilient-storage-architecture/ (accessed November 28, 2014).

35. What is the default contribution folder location; https://community.symform.com/entries/97070216-What-is-the-default-contribution-folder-location-/ (accessed February 25, 2015).

36. Birk D, Wegener C. Technical issues of forensic investigations in cloud computing environments. Proceedings of the Sixth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE); 2011 May 26; Oakland, CA. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2011.

37. Damshenas M, Dehghantanha A, Mahmoud R, Bin Shamsuddin S. Forensics investigation challenges in cloud computing environments. Proceedings of the 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec); 2012 Jun 26-28; Kuala Lumpur, Malaysia. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2012;190–4.

38. Daryabar F, Dehghantanha A, Udzir NI. A review on impacts of cloud computing on digital forensics. International Journal of Cyber-Security and Digital Forensics (IJCSDF) 2013;2(2):77–94.

39. Mason S, George E. Digital evidence and cloud computing. Computer Law and Security Review 2011;27(5):524–8.

40. Ruan K, Carthy J, Kechadi T, Crosbie M. Cloud forensics. In: Peterson G, Shenoi S, editors. Advances in digital forensics VII. Heidelberg/Berlin, Germany: Springer Berlin Heidelberg, 2011;35–46.

41. Simou S, Kalloniatis C, Kavakli E, Gritzalis S. Cloud forensics: identifying the major issues and challenges. In: Jarke M, Mylopoulos J, Quix C, Rolland C, Manolopoulos Y, Mouratidis H, Horkoff J, editors. Advanced information systems engineering. Cham, Switzerland: Springer International Publishing, 2014;271–28.

42. Marty R. Cloud application logging for forensics. Proceedings of the 2011 ACM Symposium on Applied Computing; 2011 Mar 21-24; TaiChung, Taiwan. New York, NY: Association for Computing Machinery, 2011;178–84.

43. Ab Rahman NH, Cahyani NDW, Choo KKR. Cloud incident handling and forensic-by-design: Cloud storage as a case study. Concurrency and Computation: Practice and Experience 2016; DOI: 10.1002/cpe.3868.

44. Cahyani NDW, Martini B, Choo KKR, Al-Azhar MH. Forensic data acquisition from cloud-of-things devices: Windows smartphones as a case study. Concurrency and Computation: Practice and Experience 2016; DOI: 10.1002/cpe.3855.

45. Zawoad S, Hasan R. Cloud forensics: a meta-study of challenges, approaches, and open problems. Ithaca, NY: Cornell University Library, 2013;arXiv:1302.6312[csDC].

46. Dykstra J, Sherman AT. Design and implementation of FROST: digital forensic tools for the OpenStack cloud computing platform. Digital Investigation 2013 Aug 31;10:S87–95.

47. Zawoad S, Dutta AK, Hasan R. SecLaaS: secure logging-as-a-service for cloud forensics. Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security; 2013 May 8-10; Hangzhou, China. New York, NY: Association for Computing Machinery, 2013;219–30.

48. Gebhardt T, Reiser HP. Network forensics for cloud computing. In: Dowling J, Taiani F, editors. Distributed applications and interoperable systems. Heidelberg/Berlin, Germany: Springer Berlin Heidelberg, 2013;29–42.

49. Dykstra J, Sherman AT. Acquiring forensic evidence from infrastructure-as-a-service cloud computing: exploring and evaluating tools, trust, and techniques. Digital Investigation 2012;9(Supplement):S90–8.

50. Thethi N, Keane A. Digital forensics investigations in the cloud. Proceedings of the 2014 IEEE International Advance Computing Conference (IACC 2014); 2014 Feb 21-22; Gurgaon, India. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2014;1475–80.

51. Martini B, Choo KK. An integrated conceptual digital forensic framework for cloud computing. Digital Investigation 2012 Nov 30;9(2):71–80.

52. What is forensic computing? australian institute of criminology; http://aic.gov.au/documents/9/C/A/%7B9CA41AE8-EADB-4BBF-9894-64E0DF87BDF7%7Dti118.pdf (accessed March 15, 2015).

53. Kent K, Chevalier S, Grance T, Dang H. Guide to integrating forensic techniques into incident response. Gaithersburg, MD: National Institute of Standards and Technology, 2006.

54. Martini B, Choo KK. Cloud storage forensics: ownCloud as a case study. Digital Investigation 2013 Dec 31;10(4):287–99.

55. Martini B, Choo KK. Distributed filesystem forensics: XtreemFS as a case study. Digital Investigation 2014 Dec 31;11(4):295–313.

56. Scanlon M, Farina J, Kechadi M. Bittorrent sync: network investigation methodology. Proceedings of the Ninth International Conference on Availability, Reliability and Security (ARES 2014); 2014 Sept 8-12; Fribourg, Switzerland. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2014;21-9.

57. Scanlon M, Farina J, Khac NAL, Kechadi T. Leveraging decentralization to extend the digital evidence acquisition window: case study on bittorrent sync. Journal of Digital Forensics, Security and Law 2014;9(2):arXiv:1409.8486 [cs.CR].

58. Do Q, Martini B, Choo KK. A forensically sound adversary model for mobile devices. PLOS ONE 2015;10(9):e0138449.

59. Do Q, Martini B, Choo KKR. Is the data on your wearable device secure? An Android Wear smartwatch case study. Software: Practice & Experience 2016; DOI: 10.1002/spe.2414.

60. Ab Rahman NH, Glisson WB, Yang Y, Choo KKR. Forensic-by-design framework for cyber-physical cloud systems. IEEE Cloud Computing 2016;3(1):50–9.

61. Wilkinson S. ACPO good practice guide for digital evidence. London, U.K.: Association of Chief Police Officers, 2011.

62. Zhang D, Adipat B. Challenges, methodologies, and issues in the usability testing of mobile applications. International Journal of Human-Computer Interaction 2005;18(3):293–308.

63. About URL security zones; https://msdn.microsoft.com/en-us/library/ms537183.aspx#internet (accessed January 13, 2015).

64. Safari's cache.db revisited; http://www.appleexaminer.com/ files/Safari_Cache.db_Revisited.pdf (accessed January 12, 2015).

65. Ab Rahman NH, Choo KKR. A survey of information security incident handling in the cloud. Computers & Security 2015;49:45–69.

66. Malin CH, Casey E, Aquilina JM. Malware forensics field guide for Linux systems: digital forensics field guides. Waltham, MA: Syngress, an imprint of Elsevier, 2014.

67. Do Q, Martini B, Looi J, Wang Y, Choo KK. Windows event forensic process. Heidelberg/Berlin, Germany: Springer Berlin Heidelberg, 2014.

68. Digital forensic sifting: Super timeline creation using log2timeline; http://digital-forensics.sans.org/blog/2011/12/07/digital-forensic-sifting-super-timeline-analysis-and-creation (accessed January 12, 2015).

69. Quick D, Tassone C, Choo KKR. Forensic analysis of windows thumbcache files. Scholarly Paper. Rochester, NY: Social Science Research Network, 2014.

70. Nirsoft. Webbrowserpassview v1.58; http://www.nirsoft.net/utils/web_%20browser_%20password.html (accessed January 20, 2015).

71. Canlar ES, Conti M, Crispo B, Di Pietro R. Windows mobile liveSD forensics. Journal of Network and Computer Applications 2013;36(2):677–84.

72. Yang TY, Dehghantanha A, Choo KR, Muda Z. Windows instant messaging app forensics: Facebook and Skype as case studies. PLoS ONE 2016;11(3):E0150300.

73. A forensic analysis of windows registry; http://www.forensicfocus.com/downloads/windows-registryquick-reference.pdf (accessed January 22, 2015).

74. Carvey H. Windows forensic analysis toolkit: advanced analysis techniques for Windows 8. Waltham, MA: Syngress, an imprint of Elsevier, 2014.

75. Carvey H. Instant messaging investigations on a live Windows XP system. Digital Investigation 2004;1(4):256–60.

76. Recentdocs; http://forensicartifacts.com/2011/02/recentdocs/ http://forensicartifacts.com/2011/02/recentdocs/ (accessed November 25, 2014).

77. File and directory linking;

https://msdn.microsoft.com/en-us/library/windows/desktop/aa364215(v=vs.85).aspx

(accessed February 12, 2015).

78. Our products;

https://www.starfieldtech.com/ (accessed February 12, 2015).

Additional information and reprint requests:

Ali Dehghantanha, Ph.D.

University of Salford

Room 205, Newton Building

Greater Manchester, Salford M5 4WT

United Kingdom

E-mail: A.Dehghantanha@salford.ac.uk

TABLE 1—*Configurations of virtual machines for Symform cloud forensics.*

VM snapshots	Details
Base-VM 1.0, 2.0, 3.0, 1.1 IE, 1.2 MF, 1.3 GC	A base VM snapshot was prepared for each OS as a control media to determine changes during each experiment with the following configurations: • Windows 8.1 Professional (Service Pack 1, 64-bit, build 9600) with 2GB RAM and 20GB hard disk (1.0). • Ubuntu 14.04.1 LTS with 1 GB RAM and 20GB hard disk (2.0). • Mac OS X Mavericks 10.9.5 with 1 GB RAM and 60GB hard disk (3.0). Then, we made three copies of the Windows-base snapshot for the web browsers and subsequently installed Microsoft Internet Explorer Version 11.0.9600.17351 (IE), Mozilla Firefox Version 13.0 (MF), and Google Chrome version 39.0.2171.99m (GC) each on a separate snapshot.
Install-VM 1.4, 2.1, 3.1	Using a duplicate copy of the base VM snapshots, we accessed the Symform download page (http://www.symform.com/download/) to download and subsequently install the Symform client application version 4.24.0.0 for each of the OS investigated.
Access-VM 1.4.1, 2.2, 3.2, 1.1.1 IE, 1.2.1 MF, 1.3.1 GC	A copy was made of the install/base VM snapshots (1.4, 2.1, 3.1, 1.1 IE, 1.2 MF, 1.3 GC) to examine the process of logging in Symform using the client/web applications. Since the Linux client application is a web GUI, the login was undertaken via the URL http://127.0.0.1:59234/ using the default Mozilla Firefox browser (version 13.0)
Upload-VM (Synchronise) 1.4.2, 2.3, 3.3	A second copy was made of the install VM snapshots (1.4, 2.1, and 3.1) to investigate the process of uploading files using the Symform desktop clients. The Enron dataset files were copied from the host machine to *C:\Sync*, */root/Sync/*, and */Users/[User Profile]/Sync/* of the Windows, Ubuntu, and Mac OS VMs respectively. Notice that file upload was not supported by the web application at the time of this research.
Uninstall-VM 1.4.2.1, 2.3.1, 3.3.1	A copy was made of the upload VM snapshots (1.4.2, 2.3, 3.3) to examine the process of uninstalling the Symform desktop clients using the "Programs and Features" (*Control Panel\All Control Panel Items\Programs and Features*) function of Windows 8.1, commands i.e., *sudo apt-get remove symform* and *sudo apt-get purge symform* on the Ubuntu client, and *sudo /Library/Application/ \ Support/Symform/scripts/uninstall* and *sudo /Library/Application\textbackslash Support/Symform/scripts/uninstall purge* on the Mac OS client.
Download-VM (Synchronise) 1.4.3, 2.4, 3.4, 1.1.2 IE, 1.2.2 MF, 1.3.2 GC	Additional copies were made of the install/base VM snapshots (1.4, 2.1, 3.1, 1.1 IE, 1.2 MF, and 1.3 GC) to examine the process of downloading files using the Symform client and web applications. We downloaded the files uploaded from the Upload-VM (1.4.2) to *C:\Download* , */root/Download/*, and */Users/[User Profile]/Download/* of the Windows, Ubuntu, and Mac OS VM snapshots.
Delete-VM (Synchronise)}\newline 1.4.3.1, 2.4.1, 3.4.1, 1.1.3 IE, 1.2.3 MF, 1.3.3 GC	An extra copy was created of the download VM snapshots (1.4.3, 2.4, 3.4, 1.1.2 IE, 1.2.2 MF, 1.3.2 GC) to assess the process of deleting files using the Symform client and web applications. No anti-forensic technique was applied to simulate a typical file-deleting situation.

TABLE 2—*Tools prepared for Symform investigation.*

Tool	Usage
FTK Imager v3.2.0.0	To create a forensic image of the *.VMDK* files.
dd v1.3.4-1	To produce a bit-for-bit image of the mobile devices' internal storage and *.VMEM* files.
emf_decrypter.py	To decrypt the iOS images for analysis.
Autopsy 3.1.1	To parse the file system, produce directory listings, as well as extracting/analysing the files, Windows registry, swap file/partition, and unallocated space of the forensic images.
HxD v1.7.7.0	To conduct keyword searches in the unstructured datasets.
Volatility 2.4	To analyse the running processes (using the *pslist* function), network statistics (using the *netscan* function), and detecting the location of a string (using the *yarascan* function) in the physical memory dumps.
SQLite Browser v3.4.0	To view the contents of SQLite database files.
Wireshark v1.10.1	To analyse network traffic.
Network Miner v1.6.1	To analyse and carve network files.
Whois command	To determine the registration information of an IP address.
Photorec 7.0	To data carve the unstructured datasets.
Nirsoft Web Browser Passview v1.58	To recover the credential details stored in web browsers.
Nirsoft cache viewer, ChromeCacheView 1.56, MozillaCacheView 1.62, IECacheView 1.53}	To analyse the web browsing caches.
BrowsingHistoryView v1.60	To analyse the web browsing history.
Thumbcacheviewer v1.0.2.7	To examine the Windows thumbnail cache.
Windows Event Viewer v1.0	To view the Windows event logs.
Console v10.10 (543)	To view the Mac-OS-specific log files (i.e., Apple System Logs).
Windows File Analyser 2.6.0.0	To analyse Windows prefetch and link files.
Plist Explorer v1.0	To examine the contents of Apple Property List (PLIST) files.
chainbreaker.py	To extract the master keys stored in the Mac OS Keychain dump.
NTFS Log Tracker	To parse and analyse the *$LogFile*, *$MFT*, and *$UsnJrnl* New Technology File System (NTFS) files.

TABLE 3—*Entries of forensic interest from symformsetup.log. Entries of this table may provide useful information for the practitioners.*

Relevance	Examples of log entries
Symform version installed	*2014-11-16 12:20:43,880Z [1] INFO App - Version: 4.24.0.0, versionLong: 4.24.0.0...*
The install path	*2014-11-16 12:28:02,915Z [1] INFO MachineConfigModel - Looking for existing config at C:\textbackslash Program Files\ Symform \ Node Service \ node.config...*
Login email addresses and times	*2014-11-16 12:28:16,275Z [10] INFO MachineConfigModel - Logging in user XXXXXXXXXX@gmail.com to URL https://control.symform.com/..*
The folder creation (initialisation) times, alongside the folder names, IDs, and full paths	*2014-11-16 12:30:43,292Z [10] INFO MachineConfigModel - Created folder with ID 2202013076695* *2014-11-16 12:30:43,292Z [10] INFO MachineConfigModel - C:\Sync : created Sync successfully.*
The node ID for the local device	*2014-11-16 12:32:41,227Z [11] INFO MachineConfigModel - Updating node 6E719584AC80609A2E42F92E7D54964646467549...*
The locally created sync folders, including the folder IDs, full paths, and node ID for the local device	*2014-11-16 12:33:37,448Z [11] INFO MachineConfigModel - Apply folder mapping 2202013076695, remote name=Sync, local path=C:\ Sync* *2014-11-16 12:33:37,448Z [11] INFO MachineConfigModel - Existing folder 2202013076695, owned by 6E719584AC80609A2E42F92E7D54964646467549: Retrieving the latest info on this folder from cloud control...* *2014-11-16 12:33:37,866Z [11] INFO MachineConfigModel - Retrieved folder 2202013076695. OwnerNodeId = 6E719584AC80609A2E42F92E7D54964646467549* *2014-11-16 12:33:37,867Z [11] INFO MachineConfigModel - Folder owned by this node*
The sync folders downloaded to the local device, along with the folder IDs, full paths, and the folder owners' node IDs	*2014-11-16 12:33:37,867Z [11] INFO MachineConfigModel - Apply folder mapping 2200428004356, remote name=Sync, local path=C:\ Download* *2014-11-16 12:33:37,867Z [11] INFO MachineConfigModel - Existing folder 2200428004356, owned by 080EAA1BE7AA722094245551DE22426AD3A22332: Retrieving the latest info on this folder from cloud control...* *2014-11-16 12:33:38,253Z [11] INFO MachineConfigModel - Retrieved folder 2200428004356. OwnerNodeId = 080EAA1BE7AA722094245551DE22426AD3A22332 \newline2014-11-16 12:33:38,253Z [11] INFO MachineConfigModel - Folder not owned by this node*

TABLE 4—*Entries of forensic interest from symformsync.log. Entries of this table may provide useful information for the practitioners.*

Relevance	Examples of log entries
Install path	*015-01-07 16:35:31,232Z [1] INFO NodeSettings - Loading configuration from 'C:\Program Files\Symform\Node Service \node.config'.*
An alternative method for collecting the node configuration property from the *node.config* file.	*015-01-07 16:40:00,617Z [8] INFO SyncHost -<node.config>*
Initialisation time of a sync folder, including the folder ID and full path	*2016-04-23 16:30:05,593Z [8] INFO SyncHost - Initializing folder 'C\ Sync' '2202013834924'.*
The sync time(s) associated with a sync folder, alongside the full path, folder ID, and read-write permission information.	*2015-01-07 16:30:06,350Z [8] INFO SyncHost - Syncing local folder 'C\ Sync' with cloud folder 2202013834924, direction: DownloadAndUpload, sort order: 0, resuming after interruption: False.*
The upload time for a sync file.	*2015-01-07 16:37:15,721Z [12] INFO SyncSessionMetricsCollector - Uploading '3111.zip' [0.973937bb511bd6438b87cb4bafa283e9.2 1.2] 100% completed*
The download time for a sync file.	*2015-01-07 16:40:20,399Z [10] INFO SyncSessionMetricsCollector - Downloading '3111.docx' [0.9c868c9100cbb0849eb4110bd15e8285.1 2.1] 100% completed*

TABLE 5—*Summary of findings.*
(R =Recoverable, P = Possibly Recoverable, N = Not Recoverable, N/A = Not Applicable).

Platform	Source of Evidence	Data artefacts found			
		Sync and file management metadata	Authentication and encryption metadata	Cloud transaction history	Synced files
Windows 8.1	Directory listings	**R**	**R**	**R**	**R**
	Registry	**R**	**N**	**P**	**N/A**
	Log files	**R**	**R**	**R**	**N/A**
	Web browser files	**P**	**P**	**R**	**P**
	Prefetch	**N/A**	**N/A**	**R**	**N/A**
	Thumbcache	**N/A**	**N/A**	**P**	**P**
	Link files	**N/A**	**N/A**	**P**	**P**
	RAM	**P**	**P**	**P**	**P**
	Pagefile.sys	**P**	**P**	**P**	**P**
	Unallocated space	**P**	**P**	**P**	**P**
Ubuntu 14.04 LTS	Directory listings	**R**	**R**	**R**	**R**
	Log files	**R**	**R**	**R**	**N/A**
	Web browser files (RDM)	**P**	**P**	**R**	**N**
	Thumbcache files	**N/A**	**N/A**	**P**	**P**
	RAM	**P**	**P**	**P**	**P**
	Swap partition	**P**	**P**	**P**	**P**
	Unallocated partition	**P**	**P**	**P**	**P**
Mac OS X Mavericks 10.9.5	Directory listings	**R**	**R**	**R**	**R**
	Log files	**R**	**R**	**R**	**N/A**
	Web browser files (RDM)	**P**	**P**	**R**	**N**
	Thumbcache files	**N/A**	**N/A**	**P**	**P**
	RAM	**P**	**P**	**P**	**P**
	Swap partition	**P**	**P**	**P**	**P**
	Unallocated partition	**P**	**P**	**P**	**P**
iOS 7.1.2	Directory listings	**R**	**R**	**R**	**P**
Android Kitkat 4.4.4	Directory listings	**R**	**R**	**N**	**P**
Network traffic		**N**	**N**	**P**	**N**

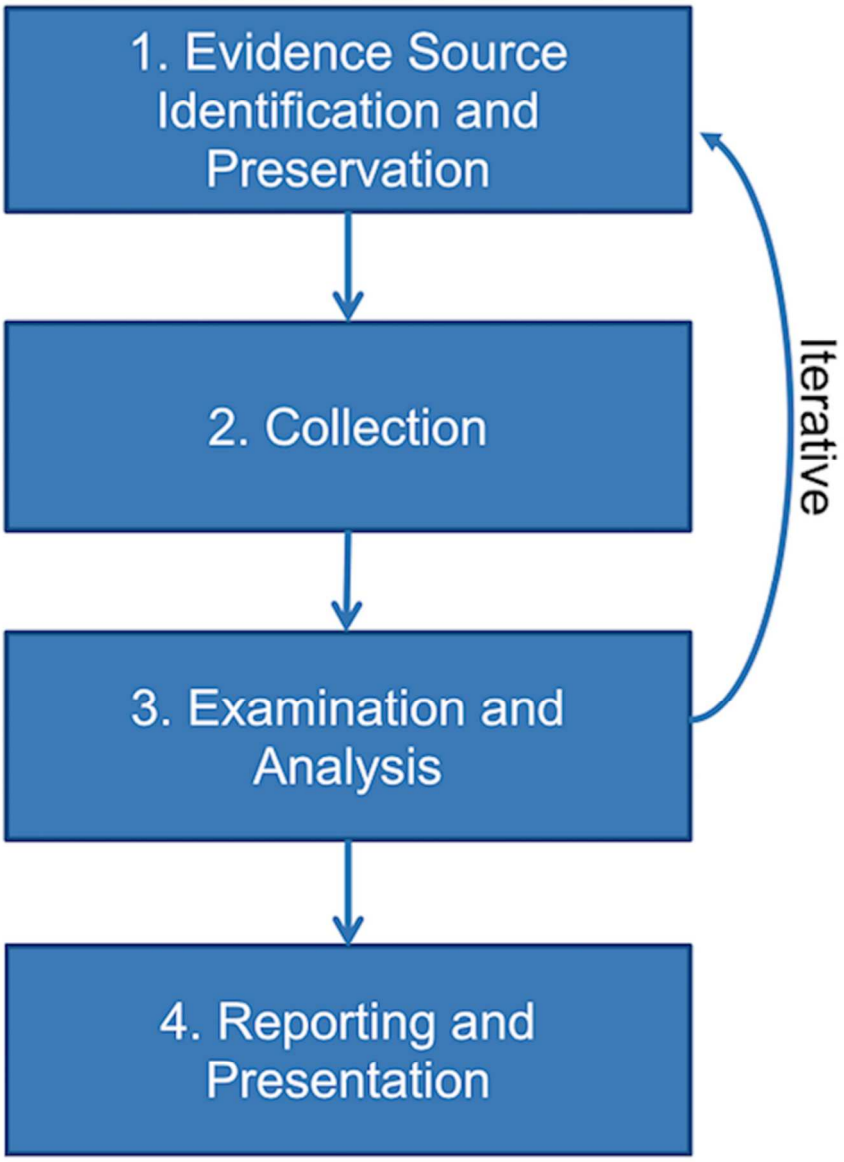

FIG. 1-Cloud forensics framework of Martini and Choo (2012).

70x94mm (300 x 300 DPI)

	Tick	ItemId	Path	PathHash	Size	ChangeReplicaId	ChangeTick	SubFolder	Tombstone	ModifiedTime	Checksum
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	BLOB	3111.docx	♦HY♦h♥@♦...	2734	2	1	0	0	2014-11-16 13:40:51Z	♦♦W♦♦V♣...
2	2	BLOB	3111.jpg	:♦) ¬♦♦♦ß♦=...	2734	2	2	0	0	2014-11-16 13:40:51Z	♦♦W♦♦V♣...
3	3	BLOB	3111.rtf	♦♦♦♦G♦ª...	2734	2	3	0	0	2014-11-16 13:40:51Z	♦♦W♦♦V♣...
4	4	BLOB	3111.txt	/♦♦y♂♦♦M...	2734	2	4	0	0	2014-11-16 13:40:51Z	♦♦W♦♦V♣...
5	5	BLOB	3111.zip	♦ſ'u♦♦_♦>♦...	2734	2	5	0	0	2014-11-16 13:40:51Z	♦♦W♦♦V♣...
6	8	BLOB	3111.pdf	♦G♦ª♦♦ka*!...	2734	2	6	0	0	2014-11-16 13:40:51Z	♦♦W♦♦V♣...
7	7	BLOB	.localized	♦♦ve▪♦P♦J...	0	2	7	0	0	2014-11-15 22:14:55Z	NULL

FIG. 2-The FolderItem table of metadata database.

170x39mm (300 x 300 DPI)

	receiver_data	request_key	time_stamp
4	BLOB	http://www.google-analytics.com/collect?non=1622650073&t=pageview&uid=106102872298592&dt=setup-account-login-create&v=1&dh=osxapp.sy...	2014-11-16 00:56:21
5	BLOB	http://www.google-analytics.com/collect?non=1115438165&t=pageview&uid=106102872298592&dt=setup-node&v=1&dh=osxapp.symform.com&dp...	2014-11-16 00:56:40
6	BLOB	http://www.google-analytics.com/collect?non=1784484492&t=pageview&uid=106102872298592&dt=setup-folders-remote-step1&v=1&dh=osxapp.sy...	2014-11-16 00:56:48
7	BLOB	http://www.google-analytics.com/collect?non=74243042&t=pageview&uid=106102872298592&dt=setup-folders-remote-step2&v=1&dh=osxapp.symf...	2014-11-16 00:57:03
8	BLOB	http://www.google-analytics.com/collect?non=114807987&t=pageview&uid=106102872298592&dt=setup-done&v=1&dh=osxapp.symform.com&dp=...	2014-11-16 00:57:18
9	{"Success": true}	http://localhost:59245/syncService/stop	2014-11-16 00:57:35
10	{"Success": true}	http://localhost:59245/syncService/start	2014-11-16 00:57:41
11	<?xml version="1.0" encoding="utf-8"?><configuratio...	http://localhost:59245/nodeConfig	2014-11-16 00:57:41
12	{"NumberOfSyncingFolders":0,"NumberOfSyncingFiles...	http://localhost:59245/metrics	2014-11-16 00:57:51
13	BLOB	http://www.google-analytics.com/collect?non=952954967&t=pageview&uid=106102872298592&dt=menulet-panel&v=1&dh=osxapp.symform.com&...	2014-11-16 00:57:51

FIG. 3-The cfurl_cache_response table of Cache.db.

170x44mm (300 x 300 DPI)

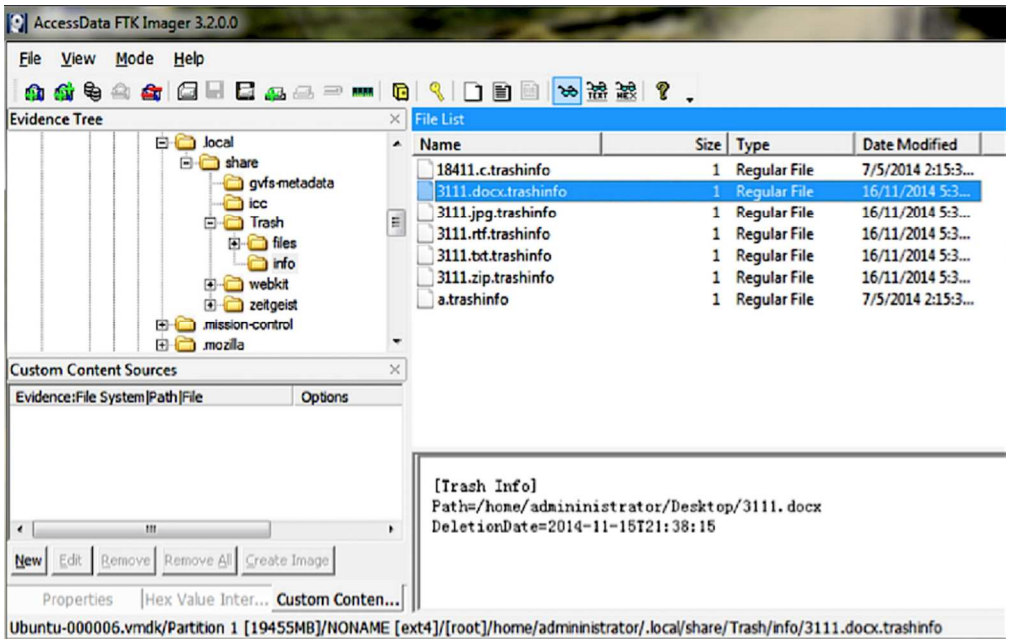

FIG. 4-Trash info for deleted files.

99x63mm (300 x 300 DPI)

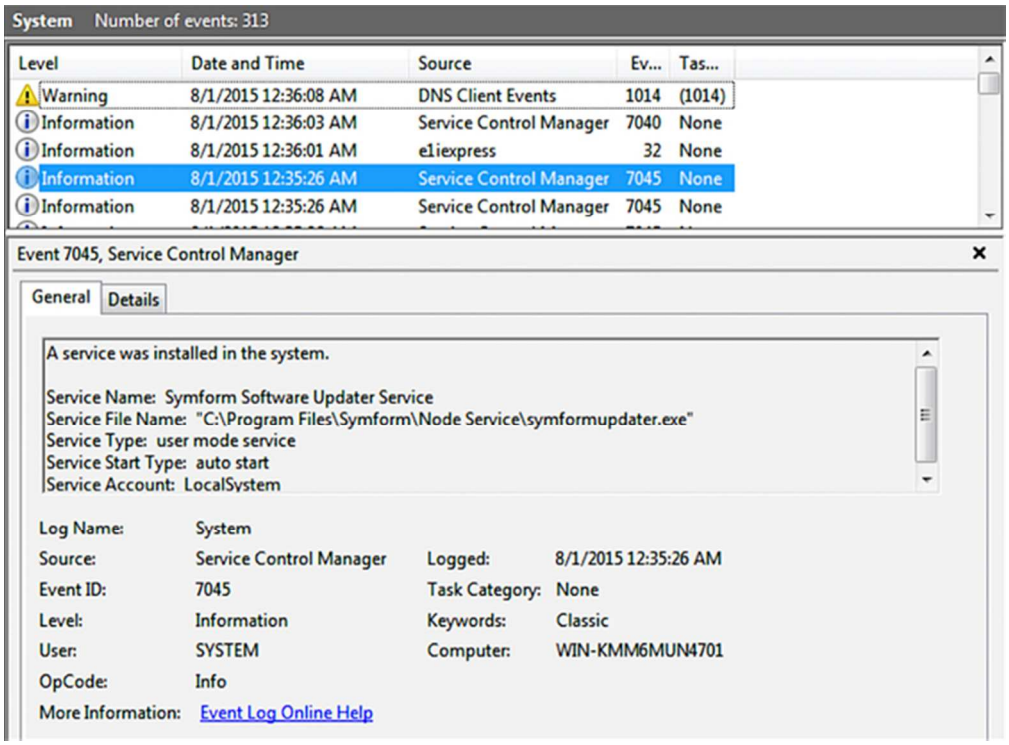

FIG. 5-Windows event log entry for Symform installation.

58x43mm (300 x 300 DPI)

FIG. 6-An excerpt of the 'pslist' output for the Symform Windows client application.

13x1mm (300 x 300 DPI)

```
Task: symformsync pid 4153 rule r1 addr 0xb65ce031
0xb65ce031  33 31 31 31 2e 7a 69 70 22 2c 22 69 73 53 75 62   3111.zip","isSub
0xb65ce041  46 6f 6c 64 65 72 22 3a 66 61 6c 73 65 2c 22 69   Folder":false,"i
0xb65ce051  73 54 6f 6d 62 73 74 6f 6e 65 22 3a 66 61 6c 73   sTombstone":fals
0xb65ce061  65 2c 22 73 69 7a 65 22 3a 32 37 33 34 2c 22 72   e,"size":2734,"r
0xb65ce071  65 64 75 6e 64 61 6e 74 53 69 7a 65 22 3a 32 37   edundantSize":27
0xb65ce081  33 36 2c 22 6d 6f 64 69 66 69 65 64 54 69 6d 65   36,"modifiedTime
0xb65ce091  55 74 63 22 3a 22 32 30 30 34 2d 31 31 2d 31 30   Utc":"2004-11-10
0xb65ce0a1  54 31 33 3a 34 30 3a 35 31 22 2c 22 69 73 52 65   T13:40:51","isRe
0xb65ce0b1  73 74 6f 72 61 62 6c 65 22 3a 66 61 6c 73 65 2c   storable":false,
0xb65ce0c1  22 69 74 65 6d 49 64 22 3a 22 30 2e 64 63 31 34   "itemId":"0.dc14
0xb65ce0d1  65 37 66 38 36 35 65 63 37 63 38 35 35 31 36 66   e7f865ec7c85516f
0xb65ce0e1  33 62 61 65 33 63 35 66 61 36 34 32 2e 35 22 2c   3bae3c5fa642.5",
0xb65ce0f1  22 63 72 65 61 74 69 6f 6e 56 65 72 73 69 6f 6e   "creationVersion
0xb65ce101  22 3a 7b 22 72 65 70 6c 69 63 61 22 3a 31 2c 22   ":{"replica":1,"
0xb65ce111  74 69 63 6b 22 3a 35 7d 2c 22 63 68 61 6e 67 65   tick":5},"change
0xb65ce121  56 65 72 73 69 6f 6e 22 3a 7b 22 72 65 70 6c 69   Version":{"repli
```

FIG. 7-Remnants of metadata database located in the memory space of 'symformsync'.

80x28mm (300 x 300 DPI)

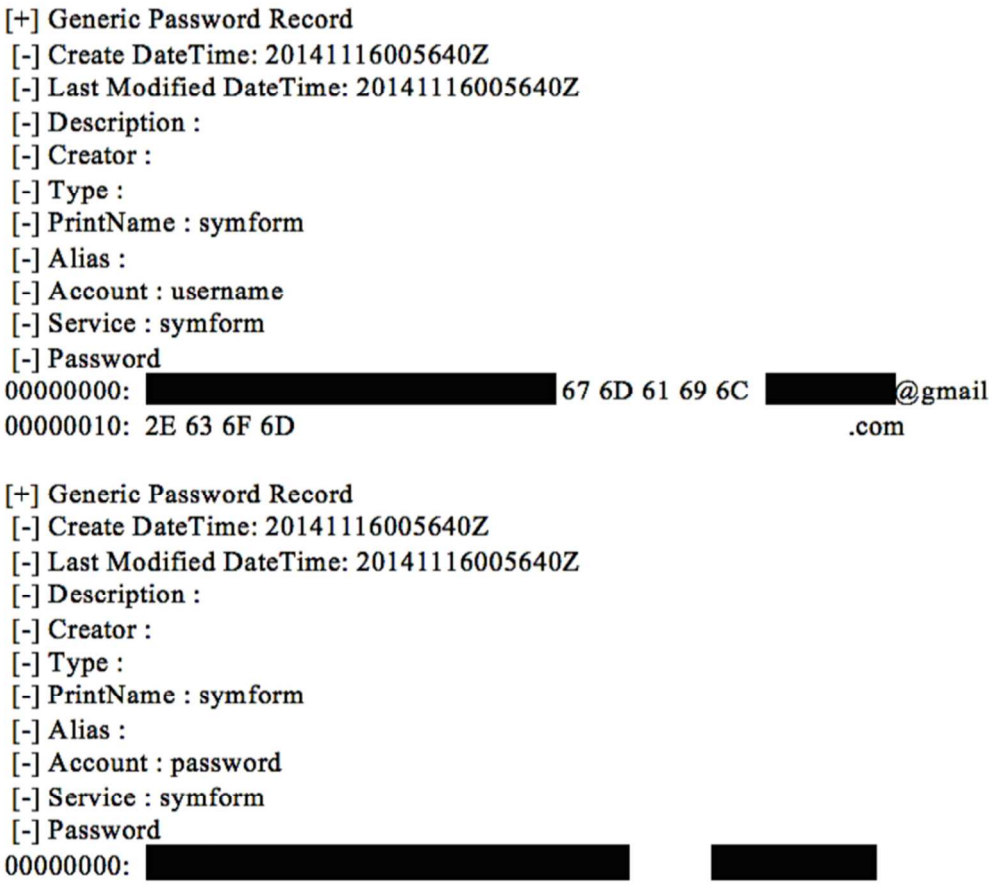

FIG. 8-An excerpt of the chainbreaker.py output.

90x81mm (300 x 300 DPI)

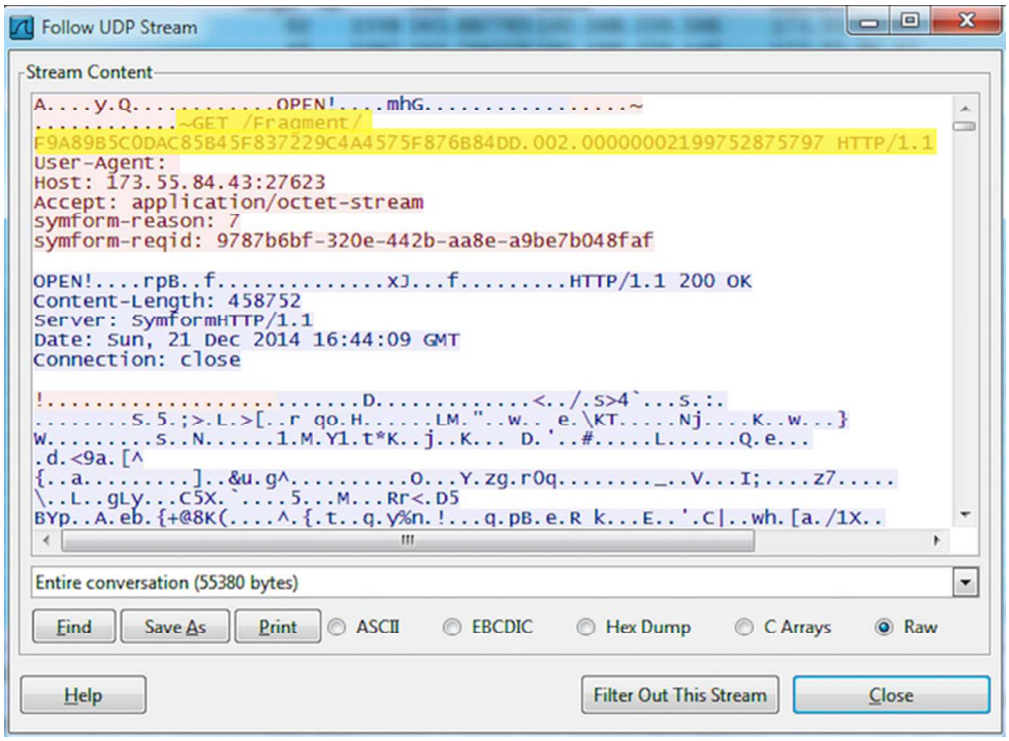

FIG. 9-An excerpt of the UDP stream containing remnants of the HTTP request for a backup file fragment.

58x42mm (300 x 300 DPI)