James Madison University

CS633 Project 1

Author:
Josh Feehs

March 5, 2014

1 Executive Summary

I was asked to come in and investigate a reported break-in to the ACME server. It was reported that the acme secrets were stolen. My investigation into the events that occurred on the ACME machine are as follows:

On September 19th, Alice, one of the users, started to teach herself how to use the Perl programming language. On September 22nd, Alice used this knowledge to create a Perl program that would copy the /acme directory (where the acme secrets were stored in encrypted and unencrypted form) into her own directory. However, she did not have the proper permissions to access the /acme directory, so she needed to find another way to copy the files. Alice did this by hiding code that created the copy in a larger package that downloads cartoons from the internet. On September 23, Bob ran that cartoon package, thusly copying the acme secrets into one of Alice's directories. From there, it was easy for Alice to steal the data when she logged in on September 25th.

My investigation also found evidence of illegal activity by Bob on the system. Bob's web history, downloads, and file system metadata show that on September 20th, Bob downloaded three illegal files from users.cs.jmu.edu/buchhofp, stored them on the server, and then hid them in other files that were copied onto the website. Although this incident is likely seperate from the stealing of the acme secrets, it is something that should certainly be reported to the police.

2 User Profiles

This section of my report will contain an initial profile of each of the users. I will describe general activities that the user has accomplished on the machine; noting what their habits and general skills are and any motives they may have had to steal the acme secrets and blackmail the company with them.

2.1 Alice

Dr. B reported that Alice was the web administrator for the company. Because of her job, I decided to first search her bash history, and then dig more into her web browsing history.

BASH HISTORY

I want to note before I go any further that the .bash_history file is something that the user can edit, so it is not necessarily a reliable source. However, I hope to see how many of the commands found in Alice's bash history seemingly had their reported results on the computer itself. Initially, Alice's bash history is very suspicious. Alice's history starts in an expected way, with Alice working in the web directories. However, right near the end of her first session (the first section before "exit"), Alice copies the entire /acme folder into /var/www, which would allow those files to be viewed from the company website. Her next session starts in her Downloads folder, where she views a file called "product_catalog.txt". This is one of the files that was in the /acme folder. Given that she was in the

Downloads folder, the folder that by default firefox sends all downloaded files to, it is possible that Alice downloaded that file off of the web server in order to test if the files were available from the internet. After this download occurs, Alice edits index.html, which could involve alloying (or disallowing) those acme files from being viewed from the web. She then ends her session by using ll on the acme folder and some of Bob's folders. Alice's third session is centered around learning and utilizing Perl. She downloaded a colection of Perl programs that grab cartoons off of websites, copies files from artwork_files to the web server, and then works on a file called test.pl. This file will be discussed in a bit more detail later. Eventually, she uses test.pl to copy the contents of the acme folder into hr test/dst/acme folder. She then copies the contents of the "test" file into a file called "stuff", then eventually gets rid of the "stuff" file using the rm command.

WEB TRENDS

I grabbed Alice's (recorded) Firefox web history from her .mozilla folder. Once I exported her history into an Excel file, I was able to look for trends in websites that she browsed. The first general trend is looking up programming tutorials. Some of Alice's first browsing session was centered around the w3schools tutorials for HTML. This makes sense per her job, as Alice is the company's web developer. Later, likely around the same time as Alice was working on her test.pl file, Alice also looked up Perl tutorials from two or three different tutorial sites. A large percentage of Alice's web history is from her gmail usage. Although the gmail URL's do not leak a lot of information, I was able to find from these that her email address is alicew.633@gmail.com, and that she downloaded at least one attachment from an email. The most interesting section of web history (personally) is Alice's financial interests. She made a web search for "get rich quick", visited many websites focused on debt relief, and also visited millionairematch.com, an online dating website for Millionaires. This implies that Alice may be in debt, but could have dreams of being significantly richer and not be content working hard for a long period of time in order to accumulate wealth.

I found one other peculiar part of her web history: a large amount of UK BBC website visits. On their own, these would not be suspicious. However, many of the same websites (in almost the same order) were found in Bob's and Dr B's web histories as well. I do not know what this means (as it may be a coincidence), but I wanted to make sure that I noted it.

POSSIBLE MOTIVES

The most obvious motive for Alice to have stolen the company secrets is monetary. Especially if she is in debt or unahppy with her financial situation (as her browsing history suggests), she would be more likely to be financially motivated to do something illegal than the average employee.

SUMMARY

Alice appears to be a technically savvy user. She learned to code in multiple languages for her job, was knowledgable enough to know how to find the .mozilla directory, and was familiar with less common commands (for a general user) like chmod g+s. This means that in my estimation, Alice has the technical means to

steal the documents, and may have the knowledge to hide some of the evidence. From Alice's web history, it appears that Alice has sufficient financial motivation to steal the files ransom as well. However, most of the data that has led me to these conclusions thus far could have been edited by Alice (or by Dr. B as the administrator), so more evidence is necessary before concluding that Alice stole the secrets.

2.2 Bob

Dr. B reported that Bob was the product designer for the company. It was also noted that he had access to the /acme directory, as that is where the product deisgns are stored. With this in mind, I decided to investigate the same areas that I investigated with Alice, as well as take a look at his general files. I made sure to pay special attention to how Bob interacted with the acme directory.

BASH HISTORY

Bob has a much more sparse bash history than Alice does. In his first session, he only opens up the products_local directory, looks at what is in it with ll, and exits. In his second session, he makes his own local copy for his producs to be stored in, goes into the acme directory, opens the acme_secrets file, then does some work on the product catalog. He runs aspell, a spell check program, on the product catalog, and then exits. His third session seems to be a bit more interesting. He creates a few image files, then combines some files (/home/bob/file1, file2, and file3) with other files he had, and then zipped up all of his product pictures.

WEB TRENDS

Bob has three main areas of interest in his web history: email, the same BBC as Alice, and pictures of cats. Bob uses hotmail as his email address. Using the key files in his .mozilla directory, I found that his email address is sideshowbob633@hotmail.com, and his password to that account is kittens633. However, when I checked online, that account no longer exists, so I was unable to access any of his emails to search for more evidence. Near the end of his web history, he had a similar selection of BBC websites as Alice did, consisting of many of the RSS feed pages for the different BBC areas. A vast majority of Bob's internet searches were of pictures of kittens. He searched both for "kittens" and "cute kittens" and went to many of the links that his google searches resulted in.

At this stage of the investigation, I found evidence that Bob has viewed, downloaded, and stored contraband on the ACME computer. His web history shows that he viewed "kittyporn1.jpg", "kittyporn2.jpg", and "kittyporn3.jpg", known contraband, at https://users.cs.jmu.edu/buchhofp/forensics, and his downloads file shows that he downloaded each of them and named them file1, file2, and file3 accordingly. These are the files that were combined with the cat command in his bash history. I investigated left_logo.jpg and right_logo.jpg, and found that right after a JPG footer (FF D9), there was another JPG header (FF D8.) This confirms that there are multiple pictures stored within those two JPG files. I did not attempt to extract the images, as they are contraband

and my possessing them would have been illegal. I advise that you pass this information on to the local police for them to investigate further. The contraband is highly likely to be found in left_logo.jpg and right_logo.jpg within artwork_files.zip.

POSSIBLE MOTIVES

Finances are always a motivating factor. Because the files are being held for a high ransom price, I do not think any one can be ruled out as not having some financial motive. However, I do not think that Bob would be nearly as likely to have financial motives as Alice would. In fact, I would expect that Bob would not want to do anything that triggered an investigation into the files on the computer, as it could lead tothe discovery of his contraband. Were Bob to think of stealing the acme secrets, I would have expected him to use them as blackmail. In the event that someone discoered his kitty porn, he could have used the acme secrets as leverage to try to coerce Dr. B into not turning him in to the police. It would not have been very advantageous for Bob to let Dr. B know that the files had been stolen.

SUMMARY

Bob appears to have a working knowledge of how to use a computer, but does not seem to be particularly technically savvy. He stored his email login and password in Firefox, used generally basic commands in the command line, and was foolish enough to download contraband onto a company computer that he shared with two others. Given the evidence I have found, I believe that it would not have been in Bob's best interests to blackmail the company for money with the acme secrets had he stolen them.

2.3 Dr B

Although Dr. B was the one to call me in to investigate the breach, it would have been rresponsible for me to not investigate him as well. He noted that he was the administrator for the acme machine, but otherwise his profile was fairly sparse. Given that he was the administrator and could gain root access under the root account, his personal logs are not as likely to be trusted as logs in Bob's profile (as Dr. B is more technically savvy than Bob.)

BASH HISTORY

Dr. B's bash history looks almost exactly as you would expect an administrator's admin to look. He installed some software, worked on seeting up the workspace for he and Bob to work in, and created the groups that exist and ensured that permissions were as he reported them. His other main work involved creating and encrypting the acme_secrets file. From his bash history, it appears that Dr. B is new to openssl, the program he used to encrypt the acme secrets. After encrypting the acme secrets, Dr. B then ran a test program to decrpyt them (and ensure that they were recoverable), but then left this file (named test) in the acme folder with read permissions for all users. This means that he, Alice, or Bob could read the acme secrets in the clear simply by opening /acme/test.

WEB TRENDS

Dr. B had a very small web history, which could have been the result of secret borwsing being used, or little web usage in general. His web history contained basic Ubuntu and Firefox pages, as well as a chunk of BBC web pages similar to the ones found in Alice's and Bob's web histories. Otherwise, his web trends do not give much insight into the habits of Dr. B.

POSSIBLE MOTIVES

As the director of ACME, Dr. B has the most to lose if the secrets are published. However, if the ransom is paid, he would get the substantial amount of money and still be in charge of a (still profitable) company. He is the one that got the computer set-up for my investigation, and as the one person that I know would have root access, could manipulate the logs on the system to say virtually whatever he wanted. He very easily has the means to carry out the theft (and subsequent cover-up), and because it is a lot of money at stake, has some level of motive. However, as clearly stated, he has significantly more to lose if the ransom is not paid and the secrets are released, as he is the director of the company (versus just being an employee.)

3 Timeline

This section will contain a high level timeline for events that took place on the system from September 19 until the system was turned out. I selected this time period as it appears that this is the time period that the malicious activity revolves around. The supporting evidence for this timeline is provided in Appendix B, sorted by date. The events are sorted in approximate order chronologically. Most events have a specific time-stamp (as well as the date) associated with it, and those that do not happen are bounded by the time-stamped ones on either side (as the timeline is chronologically ordered.)

3.1 High level timeline

Time	Event Description
9/19/11 18:24	Alice logs in to the server
9/19/11 18:24	Alice browses web- gmail, Perl study
9/19/11 18:52	Alice downloads zzamboni cartoon grab package, extracts it
9/19/11	Alice views permissions of Bob's Downloads and .mozilla folders
9/19/11 19:07	Alice logs off
9/20/11 18:53	Dr B logs in to the server
9/20/11 18:58	Dr B logs off
9/20/11 19:29	Bob logs in to the server
9/20/11 19:29	Bob browses the internet
9/20/11 19:42	Bob downloads contraband
9/20/11 19:45	Bob hides contraband
9/20/11 19:46	Bob zips artwork file
9/20/11 19:51	Bob logs off
9/21/11 18:19	Alice logs in to the server
9/21/11 18:20	Alice accesses email, retrieves artwork file
9/21/11	Alice works on the web server
9/21/11	Time unaccounted for here, possibly when Alice accessed Bob's firefox info
9/21/11 20:46	Alice logs off
9/22/11 17:25	Alice logs in to the server
9/22/11 17:26	Alice browses web and Perl specific questions
9/22/11	Alice works on the web site
9/22/11 18:36	Alice creates test.pl and its input
9/22/11	Alice tries to run test.pl on the acme directory- it doesnt work
9/22/11 18:52	Alice hides test.pl in grabcartoons.pl
9/22/11	Alice edits the destination folder- setgid bit
9/22/11 18:59	Alice zips the cartoon downloader
9/22/11 18:59	Alice logs off
9/23/11 18:59	Bob logs in to the server
9/23/11 18:59	Bob logs in to his email, reads email
9/23/11 19:00	Bob accesses the website and the cartoon grabber
9/23/11 19:02	Bob runs grabcartoons, creates acme folder
9/23/11 19:28:16	The system reboots
9/23/11 19:29	Bob logs in again
9/23/11	Bob browses the internet
9/23/11	19:35 Bob logs off
9/25/11	17:53 Alice logs in
9/25/11	17:53 Alice accesses gmail
9/25/11	17:54 Alice accesses acme folder, creates "stuff"
9/25/11	Alice uses gmail
9/25/11 17:59	Alice logs off
9/26/11 17:46	Dr B logs in
9/26/11	It appears that this is where Dr. B pulls the plug

4 Conclusions and Answers to Questions

At this point of the report, it is important for me to discuss what the data presented in the timeline could mean.

4.1 Hypothesis 1- Alice, with a bit of Bob's help

This hypothesis encorporates all of the data that I have presented thus far, and I believe is a likely explanation for the data I found.

I noted in the user profiles that Alice has ample means and motive to have stolen the ACME data. After further investigation I believe that Alice is the one who is attempting to blackmail ACME corporation. Here is the information that supports that hypothesis.

Alice's web history shows that she was looking for ways to make money as early as September 12th. Although she may have not known what she was going to do at that point, it is clear that she was looking for ways to make more money. Then, on April 19th, she looked into the acme directory and started to study Perl.This may have started as study for the company website (as the website hosts many pictures that come from the acme directory), but by September 22, she changed what she researched and wrote the test.pl file. I believe that she started working on test.pl on the 22nd because she found that she had found a specific way to get in a steal the secrets. Between the 19th and 22nd, Alice started checking up on Bob and his web downloads. I believe it was at this point that she discovered Bob's illegal activities on the company computer. When Alice realized she could not get access to the acme secrets (because of the file permissions), she decided to use Bob's access to get the files. She prepared for this on the 22nd by writing test.pl, setting up the destination directories, and setting the setgid bit on the destination directory so Bob could write to it. She then hit the code in the cartoon downloading package in order to hide Bob's activities so he was less likely to be caught and turn her in.

Bob's (coerced) involvement with this is confined to his activities on the 23rd. Bob logged in and ran the cartoon grabber, but then crashed the computer so his activity would not be logged in his bash history. He then logged back in a browsed the web in hopes of making it look like all he did was look at pictures of cats. Although his bash history does not show him running test.pl, its output is clearly attributed to him by the filesystem metadata.

Alice took it from here. She logged back in on the 25th, verified that the data had been copied, and then copied it off of the computers. Although part of the "stuff" output file was corrupted, there is residual proof of the acme files being copied on the web server (in form of a .goutputstream file, a file produced by the gnome window manager when copying files.) It is unclear if Alice copied it to the website and then copied it from there, or if she simply sent it as an attachment in an email. Regardless, the data was definitely stolen.

4.2 Hypothesis 2- Social Engineering by Alice

This hypothesis is identical to that above, with the main difference being Bob's involvement. I believe that it is possible that Bob was unaware that his actions resulted in the acme folder being copied into Alice's directories. Alice could find in Bob's Firefox history that he was interested in cartoons (like the oatmeal), and used this as a way to trick Bob into running her program. By disguising the test.pl code in the cartoon grabbing package, all Alice needed to do was trick Bob into running the package (so he could view cartoons), and he would copy the files without even knowing.

If this was the case, Bob would have had no knowledge that he was complicit in helping steal the ACME secrets. All of Alice's prep work and end work would remain the same as it was in hypothesis 1.

4.3 Answers to Questions

1. Did the trade secrets leak out? If so, who did it and how was it accomplished?

Yes, the trade secrets did leak out. As noted in my hypothesis above, Alice wrote a Perl program that Bob ran to copy the secrets, and Alice copied them out of the system over the internet. More information about how this happened is discussed above. 2. If the secrets leaked out, how could the incident have been avoided?

Dr. B did not do a good job of protecting his data. His biggest mistake was running the decryption program on the secrets and leaving the output file on disk. Because of this, Alice never had to attempt to decrypt the secrets; a process that would have been very time consuming (if not impossible.) Another mistake that Dr. B made was giving full read permissions to any user for any file in the acme directory. Dr. B seems to undertsand how file permissions work for the most part, but he could have made it so only he could even read the acme secrets (in encrypted or decrypted form.) Had he done this, Bob wold not have been able to copy out the secrets, and neither Bob nor Alice could have even opened the files without stealing Dr. B's account information (or gaining root access).

3. How should a criminal investigation now continue given the information learned from your analysis?

Two criminal investigations should continue: one for the blackmail of ACME, and one for the possession of contraband by Bob.

It appears that Alice's gmail password is stored in her GNOME keyring. Although I was not able to get access to her keyring, with more resources, the police could crack her pasword to the acme server and use that to unlock her key ring and gain access to her gmail account. I believe it is likely that there will be very encriminating evidence in her email, whether it be the secrets being emailed out or emails to Bob blackmailing him into copying the files.

Also, for the sake of understanding how to deal with Bob, the police need to investigate further in order to determine which hypothesis above regarding Bob's invovement is correct. This could be determined by questioning the involved

parties or by getting access to Bob's or Alice's email. Given that Bob will already be on the hook for his other illegal activities, I believe it is likely that he would be honest with the police.

It is clear that Bob downloaded contraband onto the acme server. The police should open an investigation into the aquisition of the contraband just like they would any other case, and can use much of the evidence I presented in this report. (See Appendix B, September 20, for more details)

5 Appendix A- Methodology and Evidence

5.1 Imaging the Hard Drive

The first step in my investigation was to image the hard drive so i could ensure that I didn't change any of its data during my investigation. I took a snapshot of the virtual machine and booted it into Helix. From there, I used dd piped into netcat (as a client) on the Helix machine, and received it on my forensics workstation using netcat (as a server). I decided to get the whole drive (sda), so that I had all of the data I could have. Had there been more space on my forensics workstation, I would have transferred over sda1 on its own as well, as sda1 is typically the actual filesystem drive. As soon as sda was copied over, I took a SHA256 hashsum of the it in order to determine what is was upon arrival. I then compared that hashsum with the hashsum on the original machine, in order to ensure that no data was corrupted while is was being transferred. As the screenshot shows, the hashsum for disk_image is the same on my forensic workstation as the hashsum for /dev/sda on the lab2. This means that disk_image is in fact a forensic duplicate of sda on the lab2 machine. There was one last step in verifying the machine- to check the times on both machines and try to determine if the times reported by the file system were real times or not. In order to check this, I ran `echo "$(date)"` on both of my virtual machines. Although I had to click from one machine over to the other, I found that they both had approximately the same times, and that the ACME machine is set in the UTC time zone.

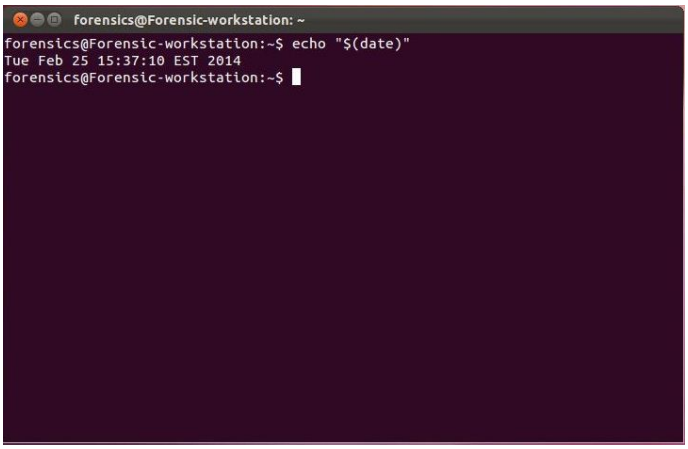


Figure 1: The time on the forensic workstation

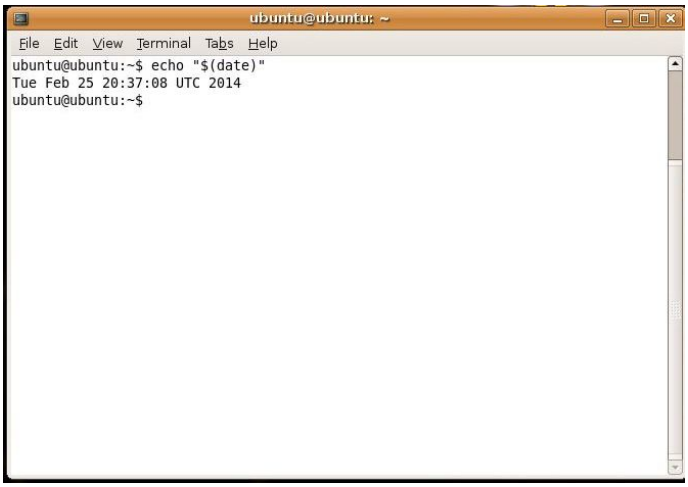


Figure 2: The time on the ACME machine

5.2 Configuring Tools

Now that I had a forensic duplicate of the hard drive, the next step was to download and configure tools to use to analyze it. For this project, we were given Sleuth Kit, a set of command line tools for forensic analysis, and Autopsy, a GUI front end fo Sleuth Kit. In order to read the ext4 file system that was on the disk image, I needed to download Sleuth Kit version 4 off of their website, `www.sleuthkit.com`. I downloaded the tar file, unzipped it, and then ran:

```
1 ./configure
2 make
```

```
3 sudo make install
```

Those three commands ensure that my environment was properly set up for Sleuth Kit, compile all of the parts of the Sleuth Kit, and install it. From there, I needed to go into my Autopsy directory and run `make` there as well. Once I had done all of that, I could run `./autopsy` from the terminal and use autopsy from Firefox at `http://localhost:9999/autopsy`.

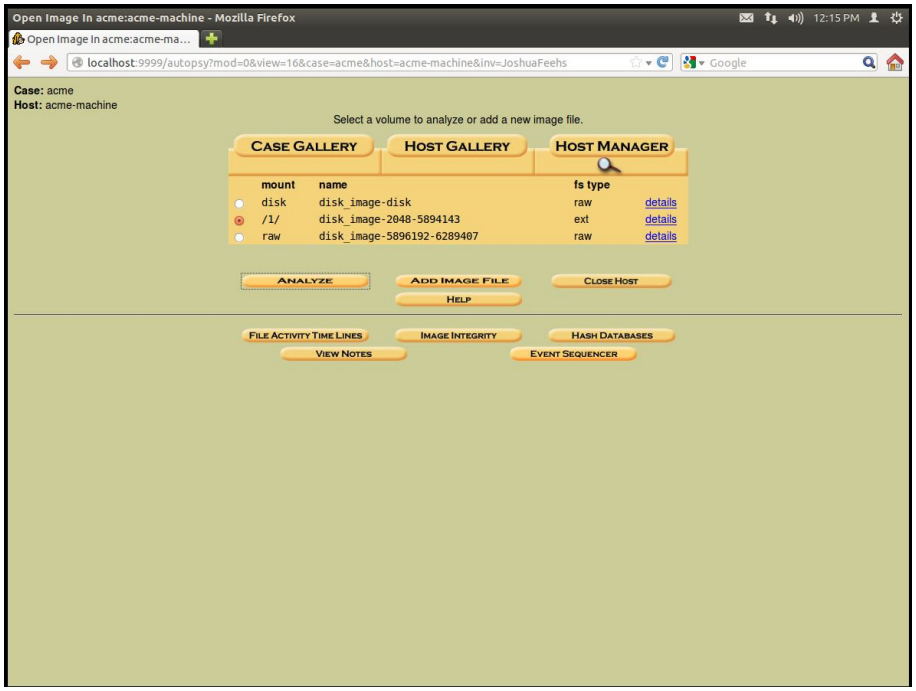


Figure 3: The main screen for autopsy once the case is set up

My next step was to set up Autopsy to use my disk image. Autopsy allows the user to start a new "case" which is the basic unit of investigation. I created a new case, making myself the host and sole investigator, and imported my disk image as the image to be investigated. As part of the setup, I had Autopsy take a MD5 checksum of the image as it imported it in order to ensure that the setup process had not changed the file. As shown IN THE PICTURE, the checksum was still the same. Now that Autopsy was up and running, I was able to start digging into the file system and see what all was on the disk.

5.3 Using Autopsy - Understanding Permissions

Given that a lot of my investigation would revolve around permissions on specific files, I wanted to try to dig into the users and groups that were on the computer

so I could better understand what the file system metadata was telling me. My first step was to check the passwd file located at /1/etc/passwd. This file contains the User ID's of each user on the machine. This is very important, as the file system metadata stores the owner of a file as that owner's UID. The passwd file shows that the root UID is 0, that Dr. B has the UID of 1000, Bob's UID is 1001, and Alice's UID is 1002.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:107::/var/run/dbus:/bin/false
avahi-autoipd:x:103:110:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:104:111:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
couchdb:x:105:113:CouchDB Administrator,,,:/var/lib/couchdb:/bin/bash
speech-dispatcher:x:106:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/sh
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
haldaemon:x:108:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
kernoops:x:109:65534:Kernel Oops Tracking Daemon,,,:/:/bin/false
pulse:x:110:115:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:111:117:RealtimeKit,,,:/proc:/bin/false
saned:x:112:118::/home/saned:/bin/false
hplip:x:113:7:HPLIP system user,,,:/var/run/hplip:/bin/false
gdm:x:114:120:Gnome Display Manager:/var/lib/gdm:/bin/false
drb:x:1000:1000:Dr. B,,,:/home/drb:/bin/bash
sshd:x:115:65534::/var/run/sshd:/usr/sbin/nologin
bob:x:1001:1001:Bob Sideshow,,,,:/home/bob:/bin/bash
alice:x:1002:1002:Alice Wonderland,,,,:/home/alice:/bin/bash
```

Figure 4: The contents of the passwd file

The next step was to look in /etc/group file in order to determine what groups exist, what their GID's are, and which users are part of which group. Of the groups in the group file, there are five main groups of interest: those with GID's from 1000-1004. Those entries in the group file are shown in the screenshot below.

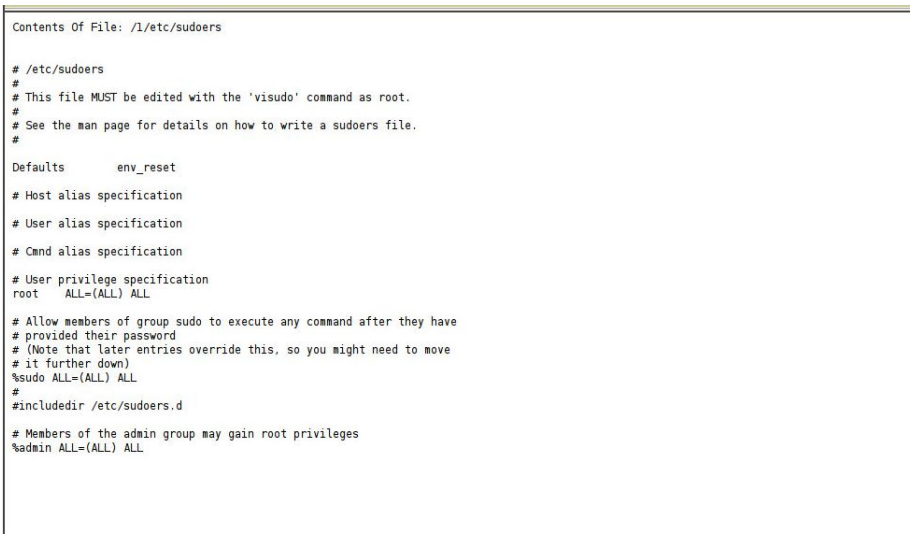
```
drb:x:1000:
sambashare:x:122:drb
bob:x:1001:
alice:x:1002:
webdev:x:1003:alice,drb
product:x:1004:bob,drb
```

Figure 5: Part of the groups file

This information will be important to the remainder of the presentation, as these numbers will help me understand the permissions on relevant files.

There was one more thing to check: which users were listed in the sudoers

file. Knowing which users could possibly issue a command as root would allow me to know who could possibly access a file as root instead of as themselves. However, when I looked at the sudoers file (/etc/sudoers), I found that only root was listed as a sudoer, so neither Alice nor Bob could issue a sudo command and temporarily act as root.



```
Contents Of File: /1/etc/sudoers


# /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# See the man page for details on how to write a sudoers file.
#

Defaults        env_reset

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL

# Allow members of group sudo to execute any command after they have
# provided their password
# (Note that later entries override this, so you might need to move
# it further down)
%sudo ALL=(ALL) ALL
#
#includedir /etc/sudoers.d

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

Figure 6: The sudoers file contents

5.4 Using Autopsy - Where are the Acme Secrets?

Now that I understood what the metadata could mean, wanted to identify where the secret files were stored on the disk and see when and how they were most recently accessed. The main acme directory was at /1/acme, where /1/ is the file system part of the disk image. This folder contained the acme secrets, as well as OTHER STUFF HERE. I verified that this file did in fact appear to be encrypted (versus being stored as plaintext), showing that Dr B's claim that the secrets were encrypted was true. However, the file permissions on the file were rw-r–r–, which means that even though Dr B is the only one that can write to it and only he and Bob are in the group that has it, any user can read the file (as seen by the last r–). There is another problem with the integrity of the ACME secrets. As seen in the screenshot below, the ACME secrets are stored in the clear in the file called "test" in the acme directory.

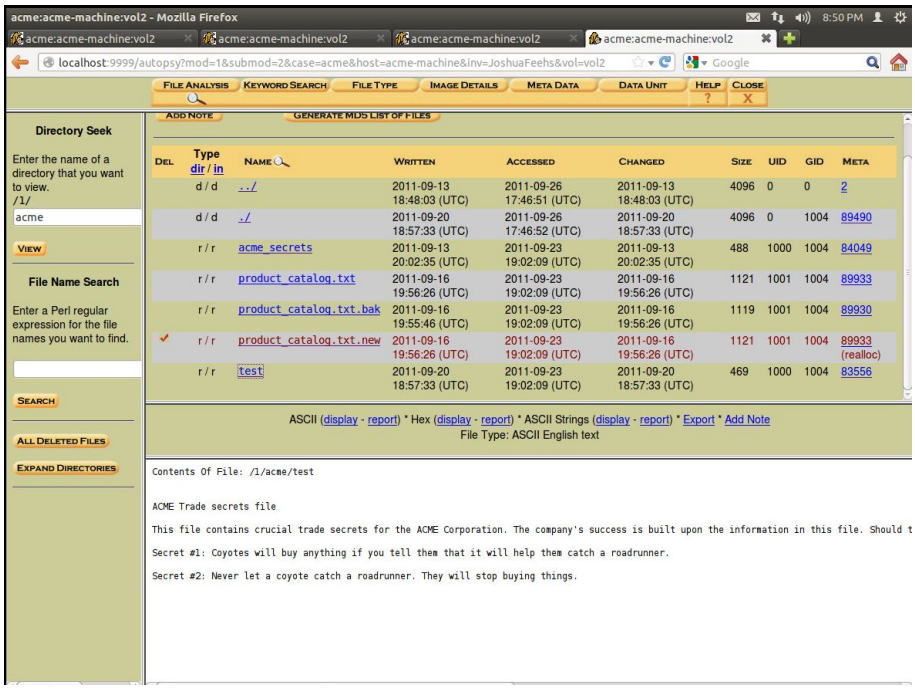


Figure 7: ACME secrets in the clear

Further investigation into this file shows that, according the Dr. B's .bash_history file, Dr. B generated this file when he tested his encryption and decryption, as shown in the screenshow below.

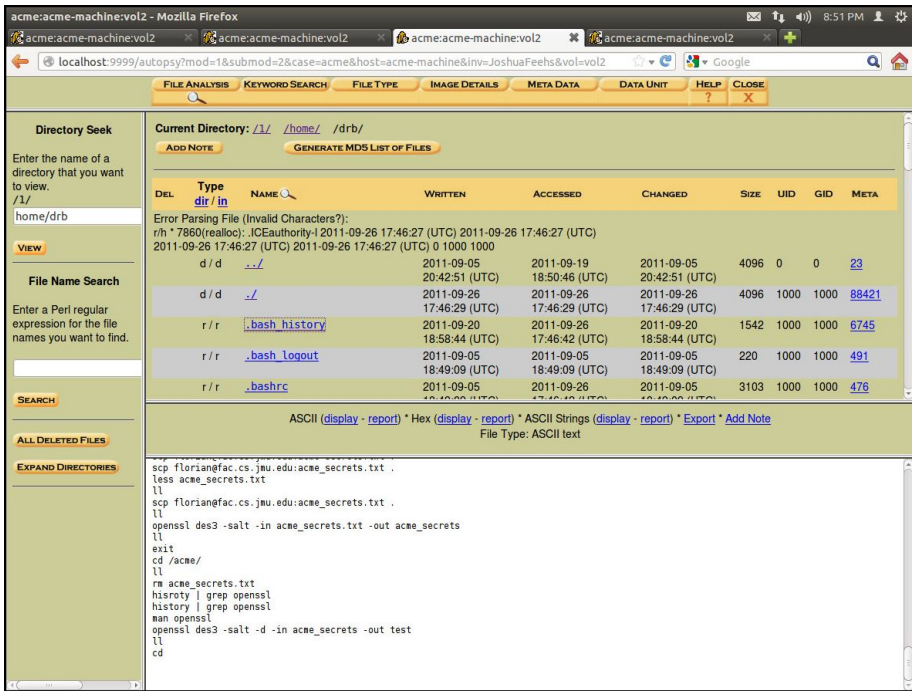


Figure 8: Dr. B creates test file

Although the existence of the test file is not suspicious, further investigation into where else the acme secrets could be stored showed some potentially nefarious behavior. According to the metadata for the acme secrets file (the encrypted version), there was once a file at /home/bob/Templates called vcmi.tar that also pointed to the same inode as acme secrets (which means that it was a link to the acme secrets file.) This could have been caused by VMWare in the copying of the files, as vcmi.tar is typically a VMWare tool file.

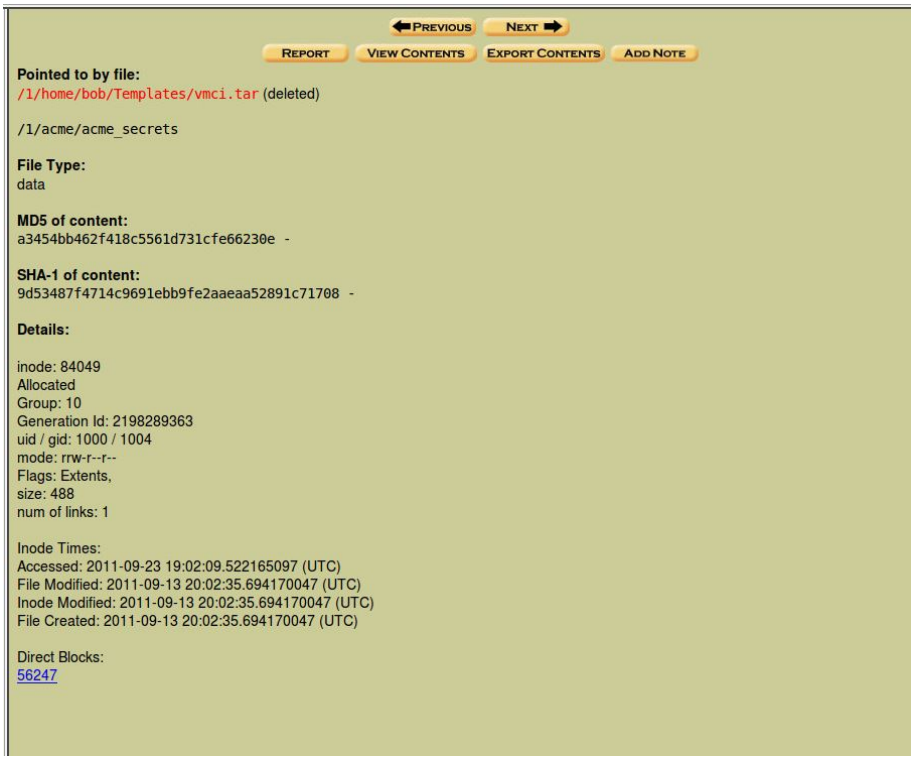


Figure 9: ACME secrets in the clear

When digging through Alice's

5.5 Firefox History

As noted in my user profiles, I went into each users' Firefox histories (located in .mozilla/user-salt/places.sqlite) to try to get to know each of the users a little better. In order to do this, I installed sqlite and sqlitebrowser so I could open the sqlite databases that Firefox uses to store the histories. I used Autopsy to export the database (places.sqlite), and then used sqlitebrowser to look through the histories. Sqlitebrowser allowed me to view the data in the database, as well as query it like a normal SQL database. This is how I acquired each of the screenshots of a users' web history.

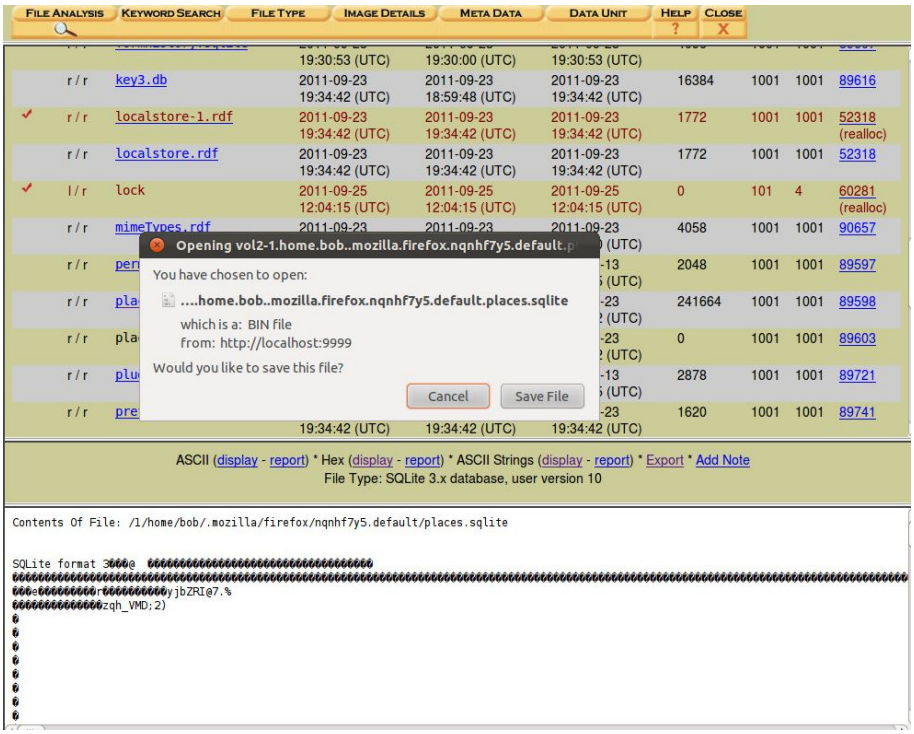


Figure 10: Screenshot of exporting a sql database

6 Appendix B - Timeline evidence

This section contains the supporting evidence for my high level timeline. The evidence has been sorted by date for easy access.

6.1 September 19

Event: Alice logs in to the server.

Evidence: The authorization log (/var/log/authlog) shows when Alice logged in. Given that there are no signs of tampering by Dr. B or the root user, this log can be trusted.

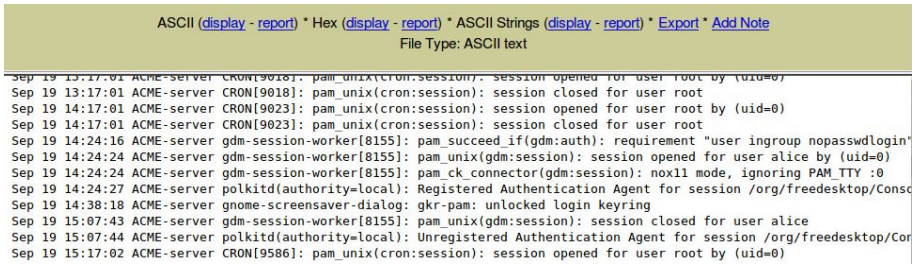


Figure 11: Screnshot of the authlog, showing that Alice logged in

Event: Alice uses her web browser to access Gmail and study Perl

Evidence: Alice's Firefox history shows that these websites were browsed. Given that there did not appear to be any programs that could have edited the Mozilla sqlite database that stores the history, it is unlikely that this was changed.

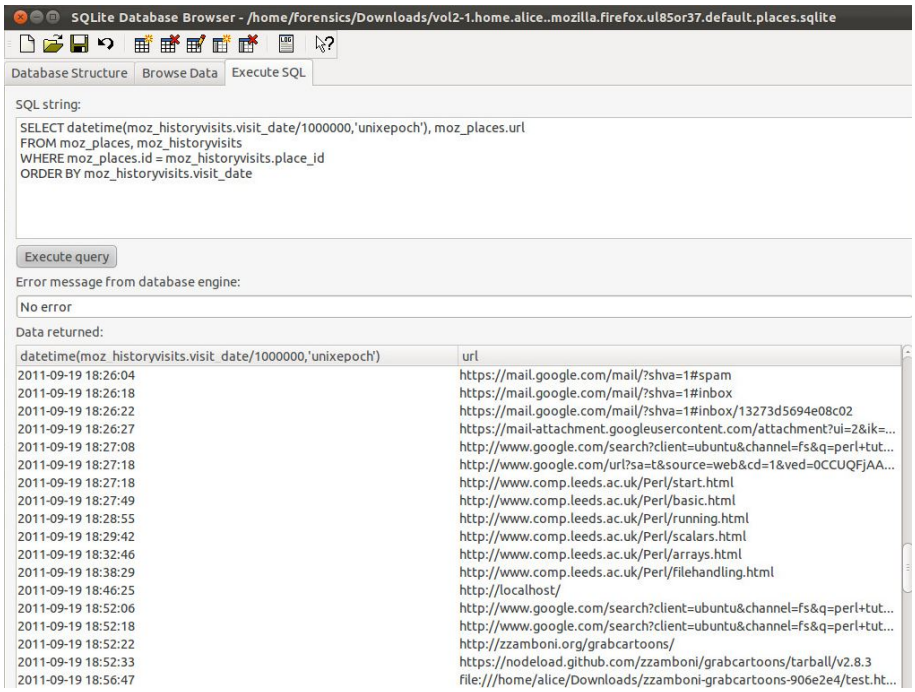


Figure 12: Screnshot of Alice's web history for 9/19

Event: Alice downloads and unzips the zzamboni catroon-grabber package.

Evidence: Alice's web history (above) shows that she downloaded the package, and the file system metadata for the unzipped (and zipped) files shows when it was created.

Figure 13: Screnshot of the metadata for the cartoon-grabber

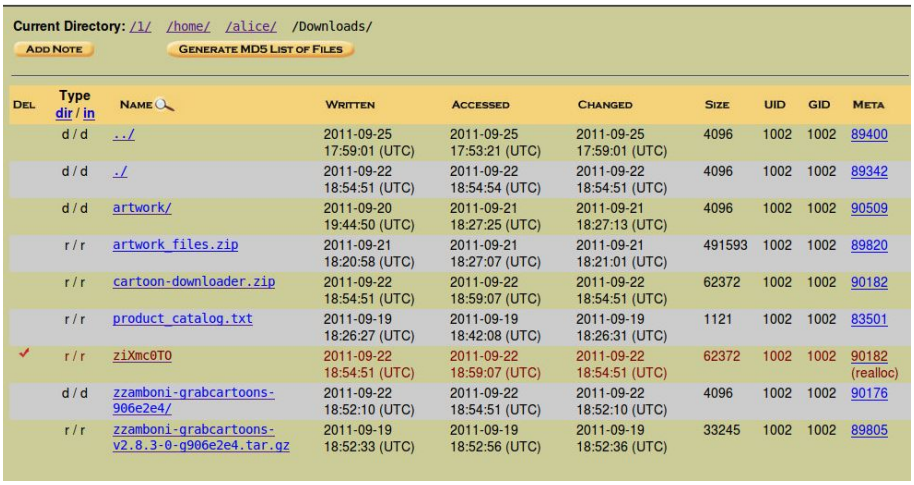


Figure 14: Screnshot of the metadata for Alice's Downloads folder

Event: Alice uses `ll` to view permissions on Bob's .mozilla/ and Downloads/ directories.

Evidence: Alice's bash history shows that she used those commands. Al-

though Alice could have edited her bash history, her bash history files does not appear to be tampered with. Every time it shows a file being edited or created, the filesystem and/or Firefox data verify what the bash history says. In this case, I was able to figure out when these permissions were downloaded by referencing the metadata of product_catalog.txt, a file that Alice downloaded during the same bash session that she viewed Bob's data. This helps give the rough estimated time when she viewed Bob's data.

Figure 15: Screnshot of the metadata for the product catalog file

The following is the section of Alice's bash history that contains when she downloaded product_catalog.txt.

```
1  cd Downloads/
2  ll
3  less product_catalog.txt
4  cd /var/www/
5  ll
6  nano index.html
7  cd
8  ll /acme/
9  ll /home/bob/
10 ll /home/bob/.mozilla/
11 ll /home/bob/Downloads/
12 exit
```

Event: Alice logs off.

Evidence: The same authlog picture that shows when Alice logged in also

shows when Alice logged off. See the screenshot that shows when Alice logged in to see when she logged off.

6.2 September 20

Events: Dr. B logs in and out of the server. This piece of evidence also shows when Bob logged in and out of the server on the 20th.

Evidence: The authlog shows when Dr. B and Bob logged in and out. Just like the earlier part of the authlog, it does not seem probable that Dr. B is corrupt, so the authlog data can be trusted.

Figure 16: Screnshot of the authlog data for September 20

Event: Bob does some general internet browsing.

Evidence: Bob's firefox history from that day shows his general web browsing: looking at mail and pictures of cats. There is no proof of Bob using any sort of SQL software, so his Firefox history is realiable, at least to the extent that he did go to the sites it shows. It does not guarantee that those are the only sites he went to, but if it is in the history, it is highly probable that he browsed that website.

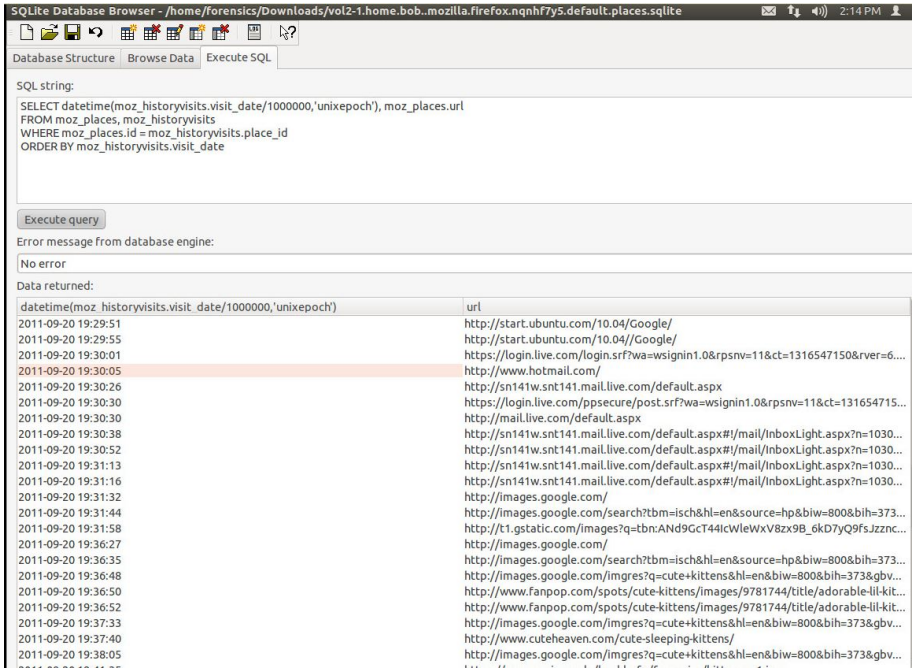


Figure 17: Screnshot of the regular part of Bob's web history for September 20

Event: Bob downloads and hides contraband.

Evidence: Firefox also stored Bob's download history. Per the other Firefox data, because Bob did not have SQL softare, he could not have added these files to his history. Once they were downloaded, Bob's bash history shows him renaming his downloaded contraband and combining it with other files. The filesystem stored metadata on those files (like left_logo.jpg). Inspection of that file with a hex editor shows a JPEG file footer immediately followed by a JPEG file header, which is likely casued by two PJPG files being concatenated together. When put together, all of this evidence presents a clear case to show that Bob did in fact download and store contraband on the company computers.

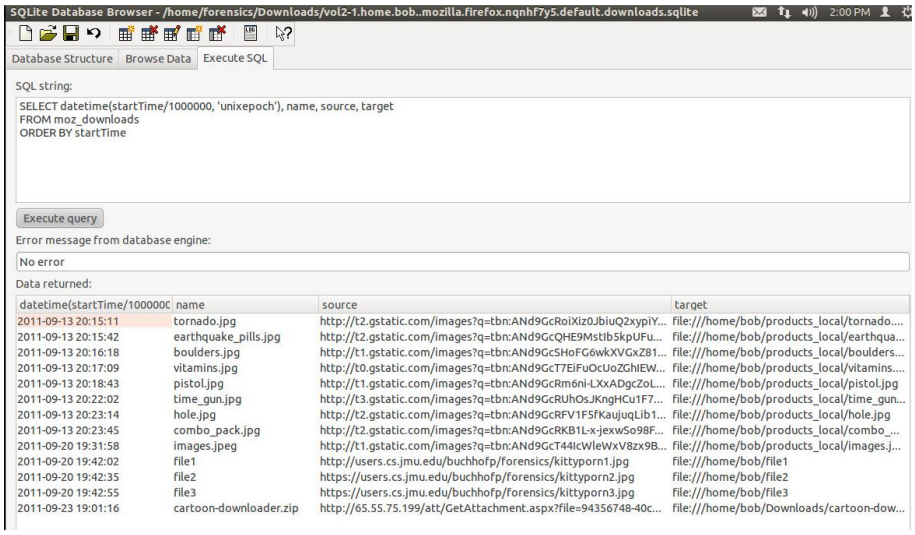


Figure 18: Screnshot of Bob's download history

```
Contents Of File: /1/home/bob/.bash_history


cd products_local/
ll
exit
mkdir proucts_local
mv proucts_local/ products_local/
cd products_local/
ll /acme/
cd products_local/
ll
cd /acme/
ll
less acme_secrets.txt
nano product_catalog.txt
aspell -c product_catalog.txt
exit
ll
cd products_local/
ll
mv images.jpeg left_logo.jpg
cp left_logo.jpg right_logo.jpg
cat /home/bob/file1 /home/bob/file3 >> left_logo.jpg
cat /home/bob/file2 >> right_logo.jpg
cd ..
mv products_local/ artwork
zip -r artwork_files artwork
ll
ls
```

Figure 19: Bob's bash history- shows file renaming

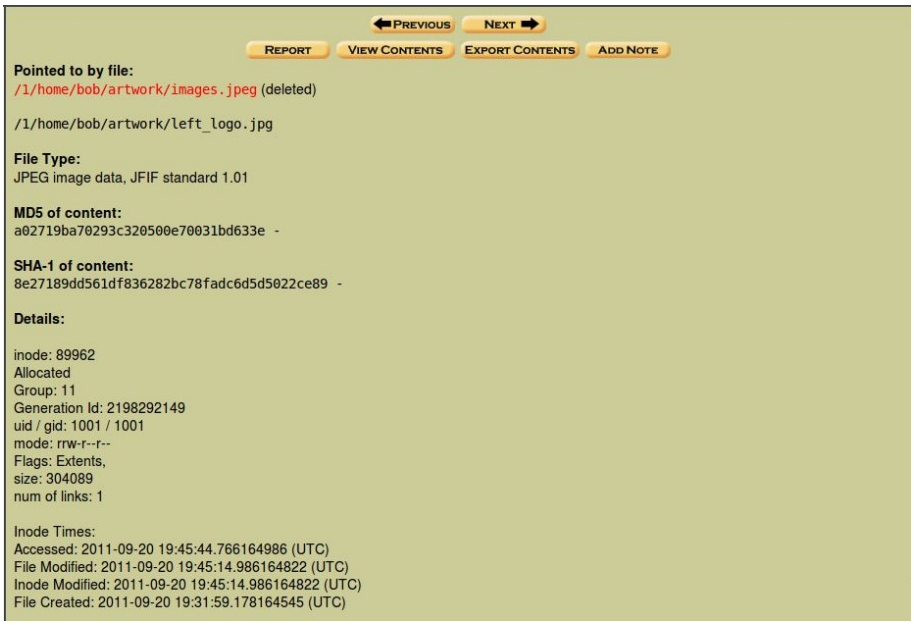


Figure 20: Metadata for the file that Bob used to hide his contraband

```
2850 1EC7 91EC 7911 2BDA 42A4 770A 06B8    (P....y.+.B.w...
A42A 4770 A06B 8A42 A477 0A06 B8A4 2A47    .*Gp.k.B.w....*G
70A0 6B8A 42A4 770A 06B8 A42A 4770 A06B    p.k.B.w....*Gp.k
8A42 A477 0A06 B8A4 7853 0E42 81A0 168E    .B.w....xS.B....
854A 3F7A 9900 ED47 47C2 EEE8 AA5A 7C89    .J?z...GG....Z|.
4BB8 AD66 DD71 19EA A825 40EC A9B8 D234    K..f.q...%@....4
A850 C86A BDA3 DA1A CCA0 0523 5960 7A64    .P.j.......#Y`zd
61EE 8C13 E307 A91D C281 AE29 0A91 DC28    a..........)...(
1AE2 90A9 1DC2 81AE 290A 91DC 281A E290    ........)...(...
A91D C281 AE29 0A3B 8504 7FFF D9FF D8FF    .....).;........
E000 104A 4649 4600 0102 0101 2C01 2C00    ...JFIF.....,.,.
00FF E11B 4B45 7869 6600 0049 492A 0008    ....KExif..II*..
0000 0008 0012 0103 0001 0000 0001 0000    ................
001A 0105 0001 0000 006E 0000 001B 0105    .........n......
0001 0000 0076 0000 0028 0103 0001 0000    .....v...(......
0002 0000 0031 0102 001E 0000 007E 0000    .....1.......~..
0032 0102 0014 0000 009C 0000 0013 0203    .2..............
```

Figure 21: Hex data- shows it is likely that the concatenation occurred

Event: Bob zips artwork

Evidence: Bob's bash history (above) shows that Bob zipped his artwork folder, and the metadata on disk verifies that it happened. This data helps put the previous events into a specific timeframe.

Figure 22: Metadata for the zipped artwork files

6.3 September 21

Event: Alice logs in to the server. Later, Alice logs out of the server

Evidence: The authlogs show when Alice logged in and out of the server. Just like the other authlogs, this is reliable.

Figure 23: Authlog showing when Alice logged in and out

Event: Alice does normal work- work on the web server and email.

Evidence: Alice's Firefox downloads show that she downloaded the artwork files that Bob created for the website, and her bash history shows that she used

his artwork while working on the website. Given the large amount of time spent in this time area, and that Bob's contraband is in the artwork folder, I believe that Alice discovered Bob's contraband during this time.

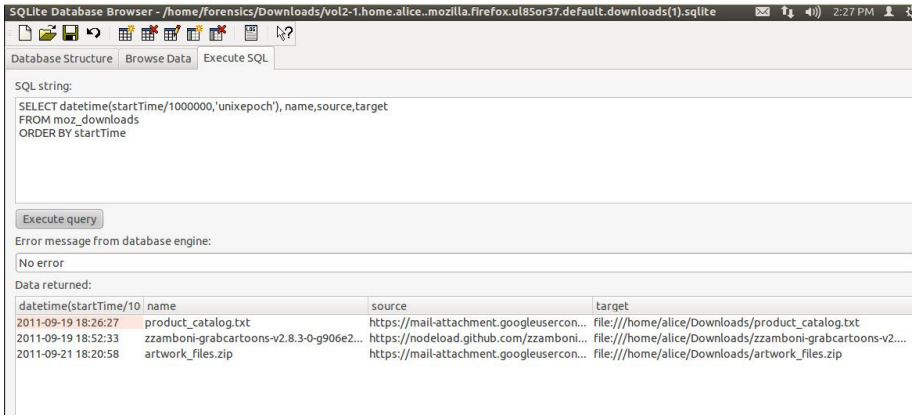


Figure 24: Alice's Firefox downloads, shows that she downloaded Bob's artwork

The following section of Alice's bash history shows how she used Bob's artwork.

```
1 cd Downloads/
2 ll
3 unzip -t artwork_files.zip
4 unzip artwork_files.zip
5 cd artwork/
6 ll
7 cp *.jpg /var/www/
8 exit
```

6.4 September 22

Event: Alice logs on to the server. Later, Alice logs off.

Evidence: Like the other auth log info, because it does not appear that Dr. B was compromised, the auth logs can be counted on.

d / d	../	2011-09-05 20:37:10 (UTC)	2011-09-21 18:27:32 (UTC)	2011-09-05 20:37:10 (UTC)	4096	0	0	12
d / d	./	2011-09-26 11:50:07 (UTC)	2011-09-26 11:50:07 (UTC)	2011-09-26 11:50:07 (UTC)	4096	0	0	81550
d / d	apache2/	2011-09-25 12:04:14 (UTC)	2011-09-26 11:50:07 (UTC)	2011-09-25 12:04:14 (UTC)	4096	0	4	89385
d / d	apparmor/	2010-03-30 19:59:44 (UTC)	2010-03-30 19:59:44 (UTC)	2011-09-05 18:47:09 (UTC)	4096	0	0	88339
d / d	apt/	2010-08-16 09:35:54 (UTC)	2010-08-16 09:35:54 (UTC)	2011-09-05 18:47:09 (UTC)	4096	0	0	88340
r / r	auth.log	2011-09-26 17:46:29 (UTC)	2011-09-25 12:04:15 (UTC)	2011-09-26 17:46:29 (UTC)	8025	101	4	81645
r / r	auth.log.1	2011-09-25	2011-09-18	2011-09-25	43408	101	4	86654

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note
File Type: ASCII text

```
Sep 22 12:17:01 ACME-server CRON[11952]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 22 12:17:01 ACME-server CRON[11952]: pam_unix(cron:session): session closed for user root
Sep 22 13:17:01 ACME-server CRON[11957]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 22 13:17:01 ACME-server CRON[11957]: pam_unix(cron:session): session closed for user root
Sep 22 13:25:29 ACME-server gdm-session-worker[11691]: pam_succeed_if(gdm:auth): requirement "user ingroup nopasswdlogir
Sep 22 13:25:36 ACME-server gdm-session-worker[11691]: pam_unix(gdm:session): session opened for user alice by (uid=0)
Sep 22 13:25:36 ACME-server gdm-session-worker[11691]: pam_ck_connector(gdm:session): nox11 mode, ignoring PAM_TTY :0
Sep 22 13:25:39 ACME-server polkitd(authority=local): Registered Authentication Agent for session /org/freedesktop/Consc
Sep 22 13:46:24 ACME-server gnome-screensaver-dialog: gkr-pam: unlocked login keyring
Sep 22 14:17:01 ACME-server CRON[12291]: pam_unix(cron:session): session opened for user root by (uid=0)
Sep 22 14:17:02 ACME-server CRON[12291]: pam_unix(cron:session): session closed for user root
Sep 22 14:17:09 ACME-server gnome-screensaver-dialog: gkr-pam: unlocked login keyring
Sep 22 14:59:53 ACME-server gdm-session-worker[11691]: pam_unix(gdm:session): session closed for user alice
Sep 22 14:59:54 ACME-server polkitd(authority=local): Unregistered Authentication Agent for session /org/freedesktop/Cor
Sep 22 15:17:01 ACME-server CRON[12673]: pam_unix(cron:session): session opened for user root by (uid=0)
```

Figure 25: The authlog data for September 22

Event: Alice browses the web and searches for Perl-specific questions

Evidence: Alice's web history shows her searching for answers to specific Perl programming questions. Like previous entries from her web history, this is fairly reliable for showing that she did browse the included websites.

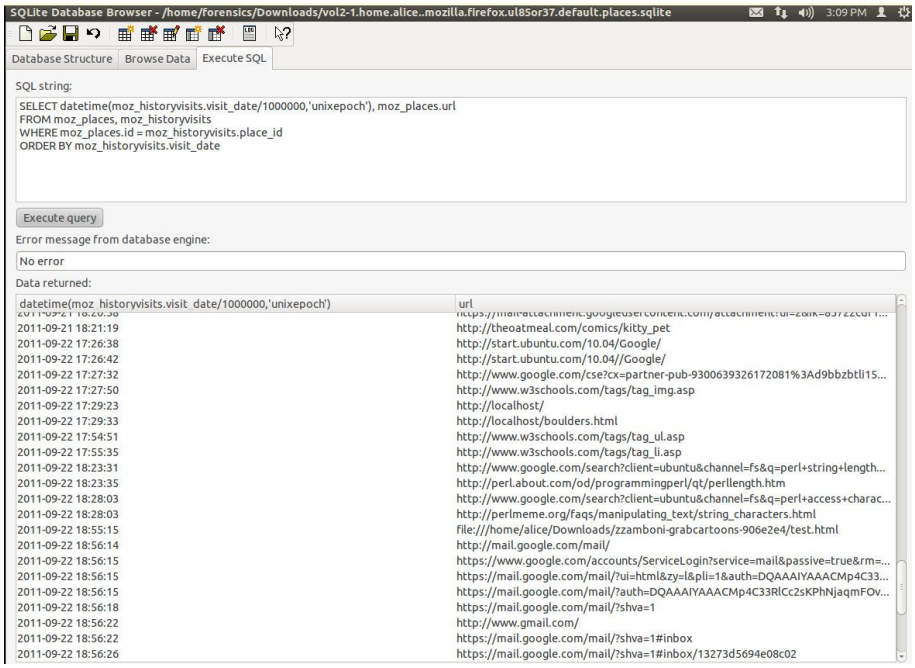


Figure 26: Alice's web history for September 22

Event: Alice works on the company website

Evidence: Alice's bash history shows that she worked on the website before she worked on test.pl. Because I have a time stamp for test.pl and the bash history is in order, it is clear that Alice worked on the website at this time. The following is the section from his bash history.

```
1 cd /var/www/
2 gedit index.html &
3 ll
4 cd
5 mkdir test
6 cd test
7 ll
8 nano test.pl
```

Event: Alice creates test.pl and its input file

Evidence: Alice's bash history (continued from the previous entries above) shows that she created the file and worked on it with nano. The filesystem metadata confirms that this occurred and gives a more accurate timestamp as to when it happened.

```
1 ll
2 nano test.pl
3 chmod u+x test.pl
4 mkdir src
5 mkdir dst
```

```
 6 less test.pl
 7 cd src/
 8 echo "This is a test" > file
 9 less file
10 cd ..
11 ./test.pl
12 ll dst/
13 ll dst/src/
14 less dst/src/file
15 rm -rf dst/src/
16 nano test.pl
17 ./test.pl
18 gedit test.pl &
19 ./test.pl
20 ll /acme/
21 ./test.pl
22 ll dst/
23 ll src/
```

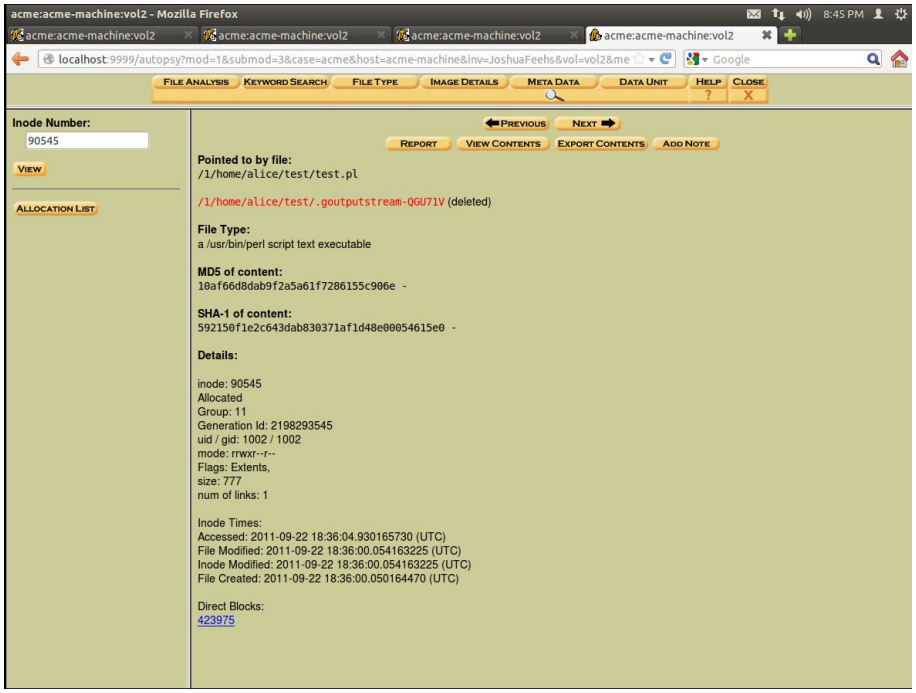


Figure 27: The metadata for test.pl

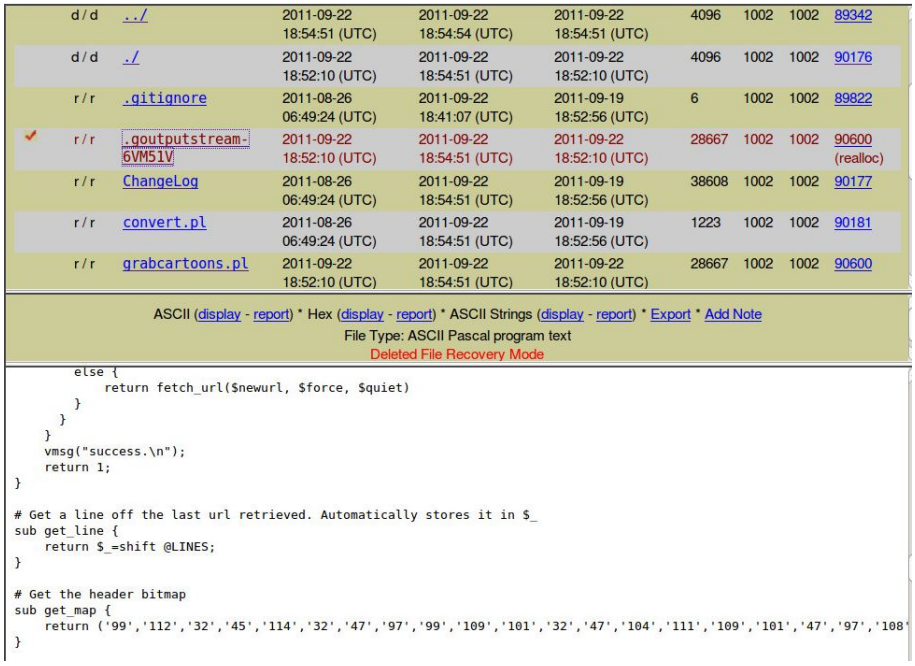
Event: Alice attempts to run test.pl on the /acme/ folder, and it doesn't work.

Evidence: Alice's bash history shows her attempting to run test.pl, and then digging into the permissions on the acme folder. This is most likely because her attempt to run the perl program didn't work.

```
1 gedit test.pl &
2 ./test.pl
3 ll /acme/
4 ./test.pl
5 ll dst/
6 ll src/
```

Event: Alice hides the code from test.pl in grabcartoons.pl.

Evidence: The filesystem metadata shows that grabcartoons.pl was edited at 18:54:51. Inspection of the code in grabcartoons.pl shows that Alice's code that she was practicing in test.pl is disguised as the method "sub get_map" in grabcartoons.pl.



	d/d	../	2011-09-22 18:54:51 (UTC)	2011-09-22 18:54:54 (UTC)	2011-09-22 18:54:51 (UTC)	4096	1002	1002	89342
	d/d	./	2011-09-22 18:52:10 (UTC)	2011-09-22 18:54:51 (UTC)	2011-09-22 18:52:10 (UTC)	4096	1002	1002	90176
	r/r	.gitignore	2011-08-26 06:49:24 (UTC)	2011-09-22 18:41:07 (UTC)	2011-09-19 18:52:56 (UTC)	6	1002	1002	89822
✔	r/r	.goutputstream-6VM51V	2011-09-22 18:52:10 (UTC)	2011-09-22 18:54:51 (UTC)	2011-09-22 18:52:10 (UTC)	28667	1002	1002	90600 (realloc)
	r/r	ChangeLog	2011-08-26 06:49:24 (UTC)	2011-09-22 18:54:51 (UTC)	2011-09-19 18:52:56 (UTC)	38608	1002	1002	90177
	r/r	convert.pl	2011-08-26 06:49:24 (UTC)	2011-09-22 18:54:51 (UTC)	2011-09-19 18:52:56 (UTC)	1223	1002	1002	90181
	r/r	grabcartoons.pl	2011-09-22 18:52:10 (UTC)	2011-09-22 18:54:51 (UTC)	2011-09-22 18:52:10 (UTC)	28667	1002	1002	90600

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note
File Type: ASCII Pascal program text
Deleted File Recovery Mode

```
        else {
            return fetch_url($newurl, $force, $quiet)
        }
      }
    }
    vmsg("success.\n");
    return 1;
}

# Get a line off the last url retrieved. Automatically stores it in $_
sub get_line {
    return $_=shift @LINES;
}

# Get the header bitmap
sub get_map {
    return ('99','112','32','45','114','32','47','97','99','109','101','32','47','104','111','109','101','47','97','108'
}
```

Figure 28: The code from test.pl in grabcartoons.pl, with metadata shown

Figure 29: This shows the line of code that is in grabcartoons.

Figure 30: The full code that was copied into grabcartoons

Event: Alice sets the setgid bit on her destination folder

Evidence: If Alice was to have Bob run the program and copy its output to her folder, she needed to setthe setgid bit on the dst folder. Her bash history clearly shows her doing this.

```
1 ll src/
2 cd ../Downloads/zzamboni-grabcartoons-906e2e4/
3 history | grep grab
```

```
./grabcartoons.pl -a -w test.html
cd
cd test/
ll dst/
less dst/src/file
groups
rm -rf dst/src/
ll dst/
ll
chmod go+w dst/
ll
chmod g+s dst/
cd ../Downloads/
```

Event: Alice zips up the cartoon downloader for upload

Evidence: Alice's bash history shows her zippping the downloader, and the file metadata supports the time stamp.

```
chmod go+w dst/
ll
chmod g+s dst/
cd ../Downloads/
ll
zip -r cartoon-downloader zzamboni-grabcartoons-906e2e4/
ll
exit
```

Current Directory: /1/ /home/ /alice/ /Downloads/

ADD NOTE GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	SIZE	UID	GID	META
	d / d	../	2011-09-25 17:59:01 (UTC)	2011-09-25 17:53:21 (UTC)	2011-09-25 17:59:01 (UTC)	4096	1002	1002	89400
	d / d	./	2011-09-22 18:54:51 (UTC)	2011-09-22 18:54:54 (UTC)	2011-09-22 18:54:51 (UTC)	4096	1002	1002	89342
	d / d	artwork/	2011-09-20 19:44:50 (UTC)	2011-09-21 18:27:25 (UTC)	2011-09-21 18:27:13 (UTC)	4096	1002	1002	90509
	r / r	artwork_files.zip	2011-09-21 18:20:58 (UTC)	2011-09-21 18:27:07 (UTC)	2011-09-21 18:21:01 (UTC)	491593	1002	1002	89820
	r / r	cartoon-downloader.zip	2011-09-22 18:54:51 (UTC)	2011-09-22 18:59:07 (UTC)	2011-09-22 18:54:51 (UTC)	62372	1002	1002	90182
	r / r	product_catalog.txt	2011-09-19 18:26:27 (UTC)	2011-09-19 18:42:08 (UTC)	2011-09-19 18:26:31 (UTC)	1121	1002	1002	83501

Figure 31: Metadata showing when the zip file was made

6.5 September 23

Event: Bob logs in. Later, Bob logs off

Evidence: The authlog shows when Bob logged in and out. Like the other authlog info, it can be trusted. This shows both times that Bob logged in on the 23rd (before and after the system rebot).

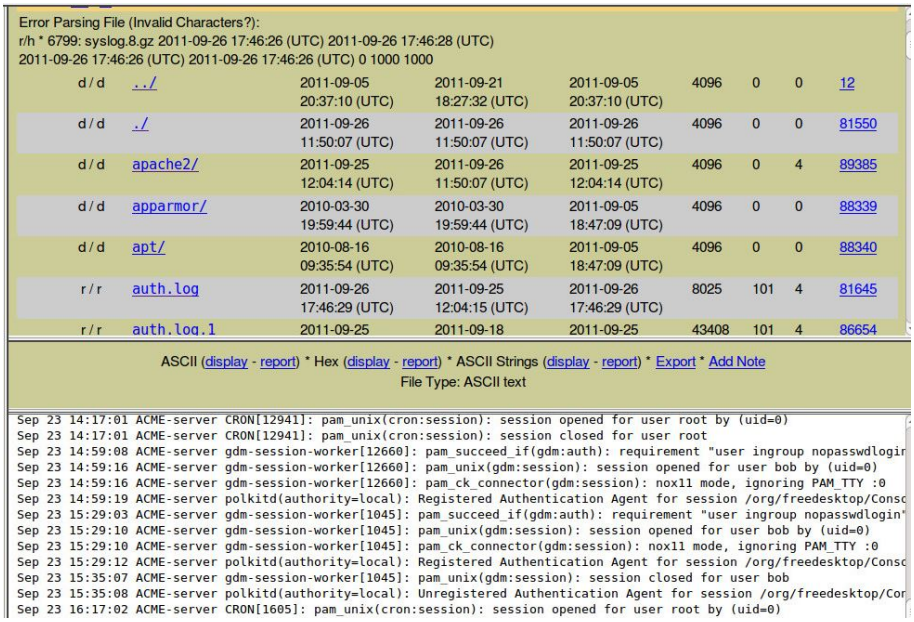


Figure 32: The full code that was copied into grabcartoons

Event: Bob uses Firefox to read email, access the company website, and access the cartoon grabber

Evidence: Bob's Firefox history shows him browsing these sites. Just like before, this data can be trusted.

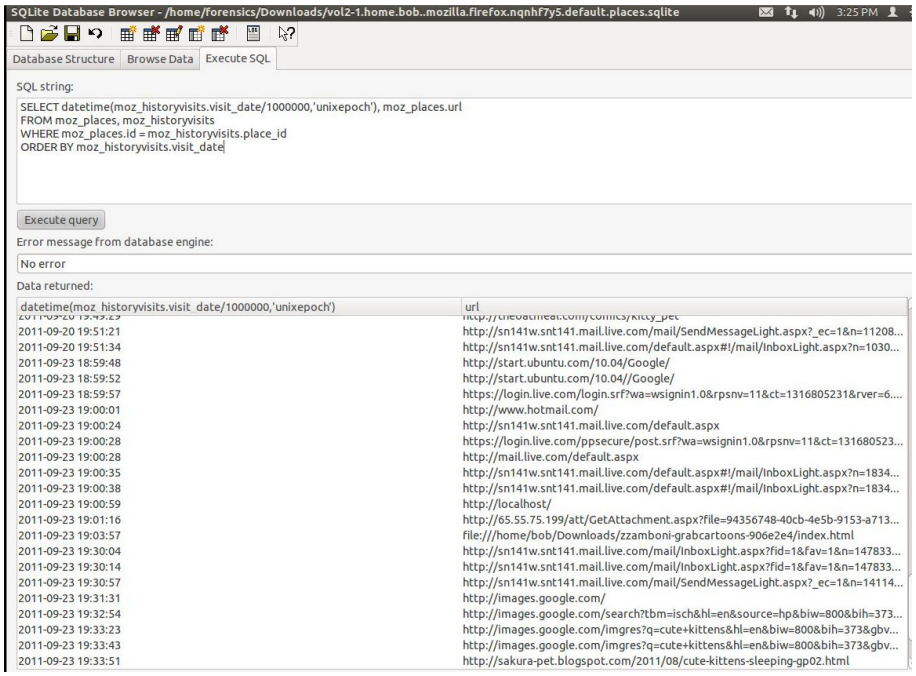


Figure 33: Bob's web history for September 23

Event: Bob runs the code that copies the acme directory

Evidence: The file system metadata shows that the acme directory in Alice's dst folder was created by the user with UID 1001 on September 23rd at the timestamp in the high level timeline. This could only have been done by Bob, who is user 1001, and who had access to code that would copy the folder over. Bob's UID can be found in Appendix A.

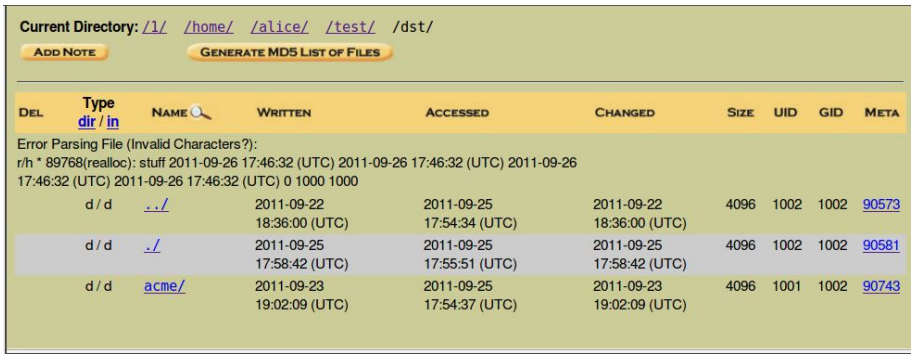


Figure 34: Metadata showing when the acme folder was created

Event: The system crashes

Evidence: It is unclear why the system crashed. However, the syslog clearly shows the system rebooting at the time noted in the timeline. The syslog is just as reliable as the authlog, as only someone with root access (not Alice or Bob) could change it.

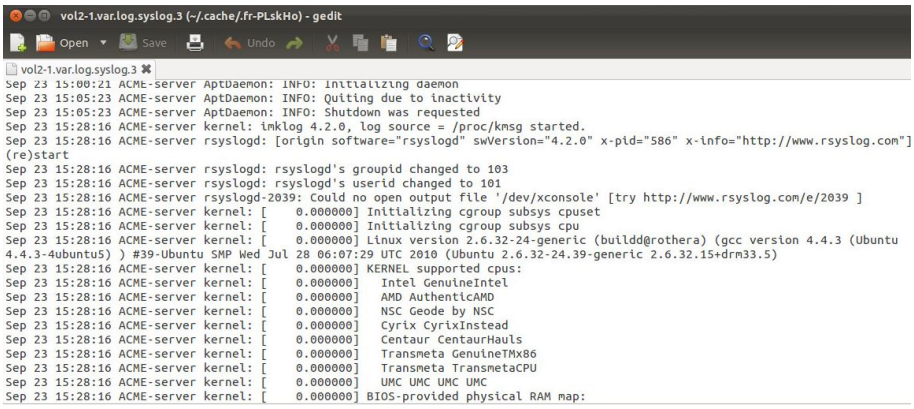


Figure 35: System logs of the reboot

Event: Bob browses the internet normally

Evidence: Bob's internet history shows him doing normal internet browsing during this time period.

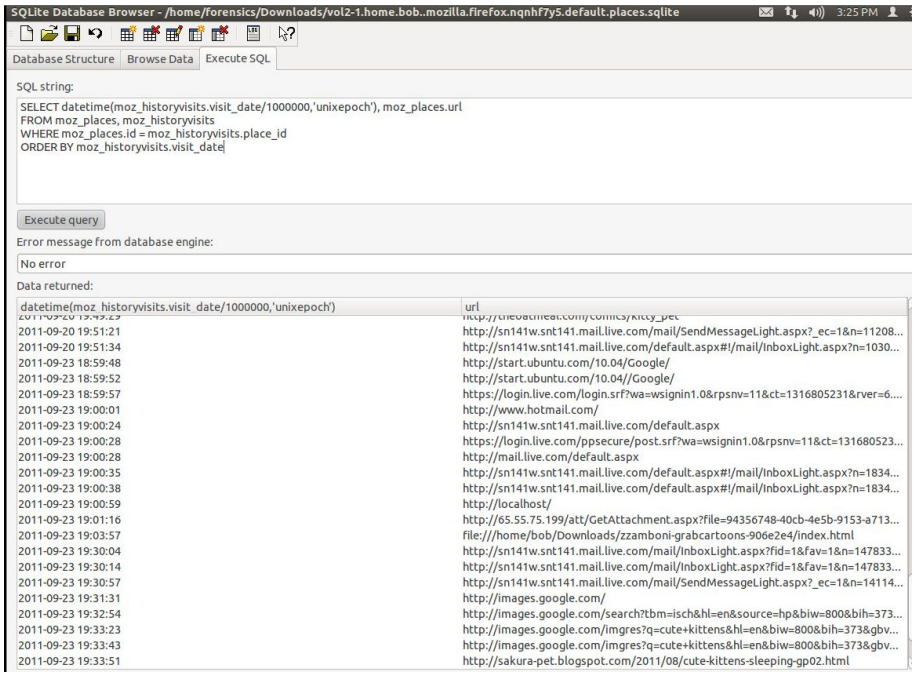


Figure 36: Bob's web history for September 23

6.6 September 25

Event: Alice logs into the server. Later, Alice logs off

Evidence: The authlogs show that Alice logged in on September 25. Just like the other auth logs, this information can be trusted to be accurate.

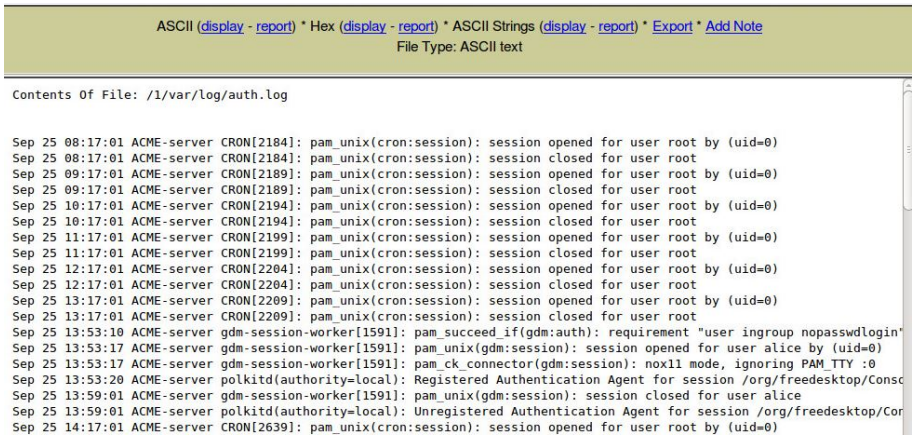
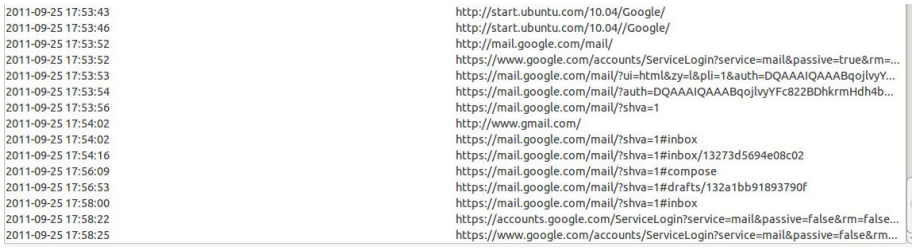


Figure 37: The authlog for September 25

Event: Alice uses gmail

Evidence: Alice's web history shows that she visited gmail multiple times on the 25th. Like the other web histories, this can be trusted.



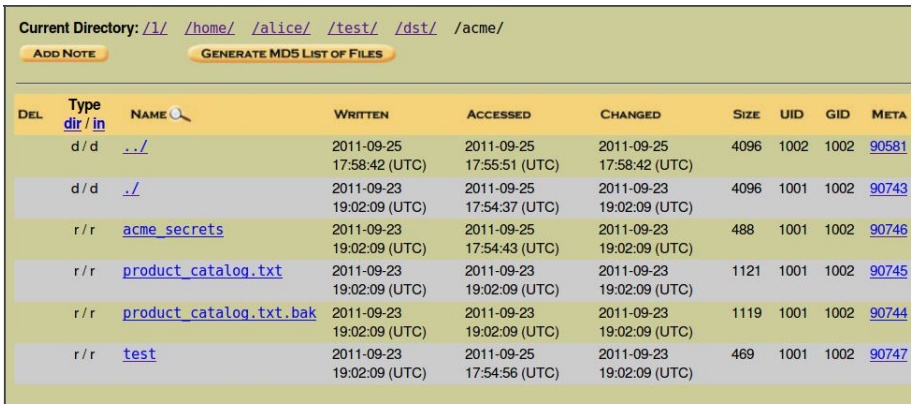
2011-09-25 17:53:43	http://start.ubuntu.com/10.04/Google/
2011-09-25 17:53:46	http://start.ubuntu.com/10.04//Google/
2011-09-25 17:53:52	http://mail.google.com/mail/
2011-09-25 17:53:52	https://www.google.com/accounts/ServiceLogin?service=mail&passive=true&rm=...
2011-09-25 17:53:53	https://mail.google.com/mail/?ui=html&zy=l&pli=1&auth=DQAAAIQAAABqojlvyY...
2011-09-25 17:53:54	https://mail.google.com/mail/?auth=DQAAAIQAAABqojlvyYFc822BDhkrmHdh4b...
2011-09-25 17:53:56	https://mail.google.com/mail/?shva=1
2011-09-25 17:54:02	http://www.gmail.com/
2011-09-25 17:54:02	https://mail.google.com/mail/?shva=1#inbox
2011-09-25 17:54:16	https://mail.google.com/mail/?shva=1#inbox/13273d5694e08c02
2011-09-25 17:56:09	https://mail.google.com/mail/?shva=1#compose
2011-09-25 17:56:53	https://mail.google.com/mail/?shva=1#drafts/132a1bb91893790f
2011-09-25 17:58:00	https://mail.google.com/mail/?shva=1#inbox
2011-09-25 17:58:22	https://accounts.google.com/ServiceLogin?service=mail&passive=false&rm=false...
2011-09-25 17:58:25	https://www.google.com/accounts/ServiceLogin?service=mail&passive=false&rm...

Figure 38: Alice's web history for September 25

Event: Alice accesses her new acme folder and creates the "stuff" file

Evidence: Alice's bash history and the metadata on the involved files show when she logged in. The metadata for "stuff" has been corrupted (as some of its blocks were written over), but autopsy can at least show that it did in fact exist at some point.

```
1  cd test/dst/acme/
2  ll
3  less acme_secrets
4  less test
5  mv test stuff
6  cp test stuff
7  cp test ../stuff
8  cd ..
9  lll
10 ll
11 rm stuff
12 exit
```

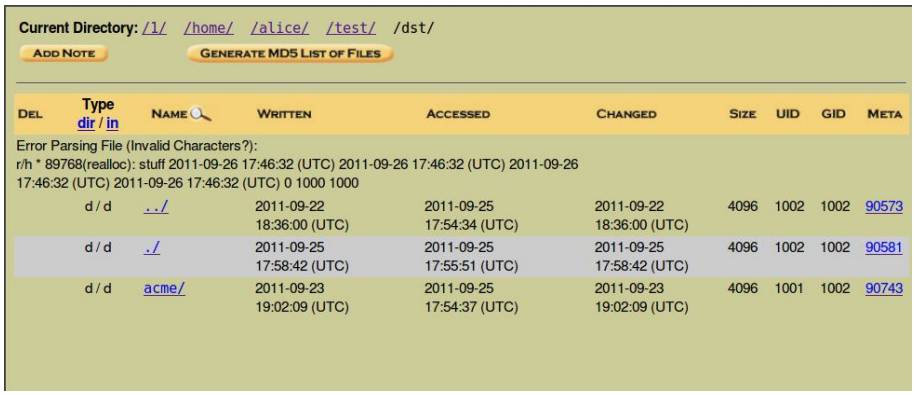


Current Directory: /1/ /home/ /alice/ /test/ /dst/ /acme/

ADD NOTE GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	SIZE	UID	GID	META
	d / d	../	2011-09-25 17:58:42 (UTC)	2011-09-25 17:55:51 (UTC)	2011-09-25 17:58:42 (UTC)	4096	1002	1002	90581
	d / d	./	2011-09-23 19:02:09 (UTC)	2011-09-25 17:54:37 (UTC)	2011-09-23 19:02:09 (UTC)	4096	1001	1002	90743
	r / r	acme_secrets	2011-09-23 19:02:09 (UTC)	2011-09-25 17:54:43 (UTC)	2011-09-23 19:02:09 (UTC)	488	1001	1002	90746
	r / r	product_catalog.txt	2011-09-23 19:02:09 (UTC)	2011-09-23 19:02:09 (UTC)	2011-09-23 19:02:09 (UTC)	1121	1001	1002	90745
	r / r	product_catalog.txt.bak	2011-09-23 19:02:09 (UTC)	2011-09-23 19:02:09 (UTC)	2011-09-23 19:02:09 (UTC)	1119	1001	1002	90744
	r / r	test	2011-09-23 19:02:09 (UTC)	2011-09-25 17:54:56 (UTC)	2011-09-23 19:02:09 (UTC)	469	1001	1002	90747

Figure 39: The metadata for Alice's acme directory that shows it was accessed when Alice was logged in



Current Directory: /1/ /home/ /alice/ /test/ /dst/

ADD NOTE GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	SIZE	UID	GID	META
Error Parsing File (Invalid Characters?): r/h * 89768(realloc): stuff 2011-09-26 17:46:32 (UTC) 2011-09-26 17:46:32 (UTC) 2011-09-26 17:46:32 (UTC) 2011-09-26 17:46:32 (UTC) 0 1000 1000									
	d / d	../	2011-09-22 18:36:00 (UTC)	2011-09-25 17:54:34 (UTC)	2011-09-22 18:36:00 (UTC)	4096	1002	1002	90573
	d / d	./	2011-09-25 17:58:42 (UTC)	2011-09-25 17:55:51 (UTC)	2011-09-25 17:58:42 (UTC)	4096	1002	1002	90581
	d / d	acme/	2011-09-23 19:02:09 (UTC)	2011-09-25 17:54:37 (UTC)	2011-09-23 19:02:09 (UTC)	4096	1001	1002	90743

Figure 40: This shows that at some point, a file named "stuff" existed

6.7 September 26

Event: Dr. B logs in. However, there is no entry that shows him logging out

Evidence: This is the last information recorded in the auth logs. I believe that this is when Dr. B logged in, received the ransom notice, and pulled the plug. The auth log shows when he logged in reliably.

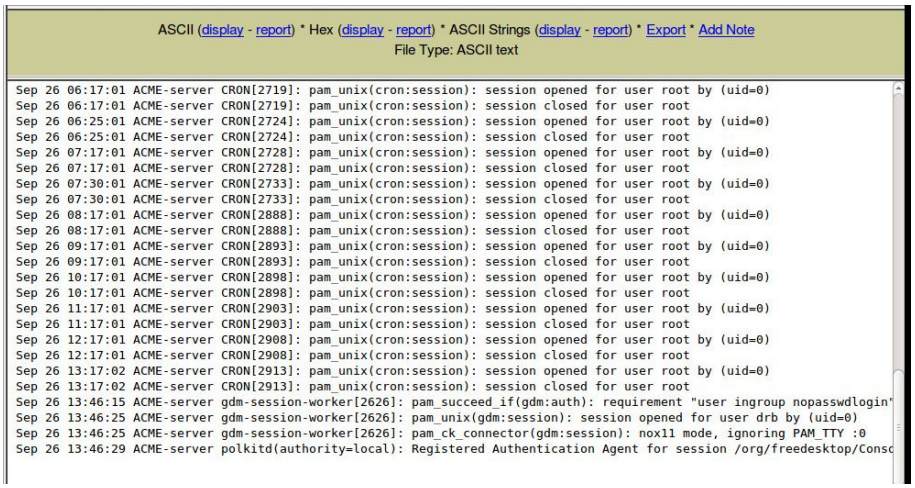


Figure 41: The authlog for September 26