EECS 398 Project 4: Forensics

Revision History

- 1.0 (December 1, 2009) – initial release.

1 Introduction

In this project, you will investigate the murder of Hapless Victim, a leading figure in the university community, while he was working in the EECS building sometime between midnight and 6 AM. While the photos of the crime scene itself have been bogged down with paperwork, the projectile pictured at `http://www.eecs.umich.edu/courses/eecs398/nerf_dart_toy.jpg` was recovered and appears, inexplicably, to have been the cause of death. The leading suspect, Nefarious Criminal, has been taken into custody, and his computer has been seized. Your job is to examine the image of his hard drive that we have made available at `http://www.eecs.umich.edu/courses/eecs398/forensics_release.vdi` and document any evidence relating to the murder.

WARNING: it may be impossible to complete this project before the deadline if you do not begin several days beforehand. Please plan accordingly.

2 Ethics

This is a forensics project, in which you will be investigating the use of computers as part of an entirely fictitious crime. You may access Web pages and other obviously-public computer services in a read-only fashion, but *you must not attempt to log into any computers or accounts over which you do not already have sole control*, even if you recover authentication credentials for those accounts from your analysis. If you think that doing so is necessary to recover additional evidence (hint, hint), you must first email the GSI for the course at `swolchok@umich.edu`. (Failure to do so is certain to result in wasted time on your part.)

3 Goals

1. Recognize how much information about the recent uses of a computer is left even in the presence of attempts to conceal it.
2. Be able to retrieve information from a disk image without booting the operating system contained on the disk.

4 Deliverables

The deliverables for this project are your answers to the following prompts. Unless otherwise indicated, your answers should be *complete* but *concise*; none of the questions should require more than 1-2 paragraphs to answer. For each prompt, include the evidence you are able to recover in its entirety, but be sure not to attach any irrelevant material or simply duplicate the contents of any files you attach in your explanation (a

penalty will apply if you ignore this direction) . Place your responses in a plain text (not DOC, RTF, PDF, or any other format) file called `evidence.txt` and indicate clearly which of the other files you've included are relevant to each response.

1. Try booting the suspect's machine and using it normally. What *specific* features of this machine make this a bad idea? We strongly recommend that you mount the suspect hard drive from a safe system before continuing (see the hints on creating a raw image later in this section).
2. What operating system does the suspect use? (be careful and specific; e.g., "Windows 2000" instead of just "Windows") (no attachment necessary)
3. What is the username of the account typically used by the suspect? (no attachment necessary)
4. Do you have any evidence that the suspect had an accomplice that was physically present on the night of the crime?
5. Were there any suspicious-looking encrypted files on the machine? If so, please attach their contents and a brief description of how you obtained the contents.
6. What evidence do you have that the suspect owned or was researching weapons of the kind involved in the murder? Please attach the specific evidence and a brief explanation.
7. Did the suspect try to delete any files before his arrest? Please attach the name(s) of the file(s) and any indication that you can find of their contents. (Hint: We will be impressed enough to grant extra credit if you manage to recover the *original* contents of a particular incriminating file, but we do not expect you to do so.)
8. Is there anything else suspicious about the machine?

Also, be on the lookout for any other machines or network services that the suspect may have used, as they may contain important evidence. Be sure to contact your supervisor as mentioned in Section 2 *before* attempting to access any such machines or accounts. Again, start early; management has been known to take up to 24 hours to respond to such requests on weekdays and 48 hours on weekends, although we try to respond promptly.

In addition to the hints we've dropped elsewhere, here is an incomplete list of some things you may want to try:

- Examine the system logs.
- Check for deleted or encrypted files.
- Search the drive image itself (e.g., using `grep -a` or `strings`) for strings that may indicate relevance to your investigation. You'll need to convert the disk image to a "raw" binary image; see the help for the "VBoxManage clonevdi" command. Also, note that a new version of VirtualBox, version 3.0.12, has been released since the previous project; we are not aware of a *requirement* to upgrade, but it may be a good idea.

Some additional resources that *may* help you:

- `http://darkdust.net/writings/diskimagesminihowto` explains how to find the partitions in a disk image and mount one of them, albeit in a different context than forensics. `http://en.wikipedia.org/wiki/Disk_partitioning` provides some background that you may be missing.
- John the Ripper (`http://www.openwall.com/john/`) is the canonical Unix password cracker. Cain and Abel (`http://www.oxid.it/cain.html`) and Hydra (`http://www.thc.org/thc-hydra/`) are two other well-known general-purpose password crackers. fcrackzip (`http://home.schmorp.de/marc/`

fcrackzip.html) is a ZIP password cracker, and pdfcrack (http://sourceforge.net/projects/pdfcrack/) is a PDF password cracker. john, fcrackzip, and pdfcrack are conveniently available in the Debian package repositories and may be available for other Linux distributions as well. When using a password cracker, it is wise to make sure that the password is not susceptible to a dictionary attack and does not use a restricted character set (e.g., lowercase letters, letters only, letters and numbers only) before spending time on a full brute-force crack. It is also a good idea to crack a very vulnerable password first to make sure you are using the tool correctly.

- "Deleted files recovery howto" (http://e2undel.sourceforge.net/recovery-howto.html) explains how to recover deleted files on the ext2 filesystem using e2undel and debugfs. http://www.xs4all.nl/~carlo17/howto/undelete_ext3.html explains how to attempt to recover deleted files on the ext3 filesystem.
- A general working knowledge of Linux is probably helpful for this project as well. If you don't have this yet, you may need to spend a little time Googling and/or experimenting to get up to speed. The GSI will also answer general Linux questions as a last resort.

5 Collaboration

You will work in groups of 2 for this project. If you wish, you may work with a different partner than you worked with for the previous projects. If you cannot find a group member, you *must* email the GSI no later than 5 PM Thursday, or we will assume that everyone has a group. If you work alone without emailing the GSI, your project will receive an automatic penalty of one letter grade in addition to any lateness penalties.

You are bound by the Honor Code not to communicate with anyone outside your group and the course staff regarding your solutions to this project. Note that searching is part of this project, so the number of pieces of evidence you have found, the techniques you have tried, how successful said techniques have been, the general progress you have made, etc. are part of your solution and should not be shared between groups.

6 Extra Hints

Given the nature of this assignment and its strict collaboration policy, HQ recognizes the need for some hints. We have developed standard hints for each question we have asked; if your group gets stuck, you may email swolchok@eecs.umich.edu with the names of your group members, the question on which you would like a hint, and the progress you have made thus far on that question. Be sure to put "EECS 398" in the subject line of your email. Each group may receive up to three hints in total, and we will enforce a 1-hour delay between hints for each group. We will respond to hint requests in a best-effort, first-come first-served fashion; in particular, you will not necessarily receive a hint before the project deadline if you request one within 24 hours of the deadline. Start early!

Note: requesting access to a remote machine does not count as a hint request, nor does asking for help with the first three questions, which are intended to help you get started.

7 Turning It In

Submit your project via email to swolchok@eecs.umich.edu by 5:00 PM on December 11, 2009. Your submission should be a gzipped tar archive named proj4.uniqname1.uniqname2.tar.gz with all the files in one directory called proj4.uniqname1.uniqname2. The subject of the email should be "EECS 398 Project 4".

Late submissions will incur a penalty of one letter grade per day *or fraction thereof*. Do not be late.