Chapter 2
Computer Forensics Education – the Open Source Approach

Ewa Huebner, Derek Bem, and Hon Cheung

Abstract In this chapter we discuss the application of the open source software tools in computer forensics education at tertiary level. We argue that open source tools are more suitable than commercial tools, as they provide the opportunity for students to gain in-depth understanding and appreciation of the computer forensic process as opposed to familiarity with one software product, however complex and multi-functional. With the access to all source programs the students become more than just the consumers of the tools as future forensic investigators. They can also examine the code, understand the relationship between the binary images and relevant data structures, and in the process gain necessary background to become the future creators of new and improved forensic software tools. As a case study we present an advanced subject, Computer Forensics Workshop, which we designed for the Bachelor's degree in computer science at the University of Western Sydney. We based all laboratory work and the main take-home project in this subject on open source software tools. We found that without exception more than one suitable tool can be found to cover each topic in the curriculum adequately. We argue that this approach prepares students better for forensic field work, as they gain confidence to use a variety of tools, not just a single product they are familiar with.

2.1 Introduction

Software products have been used in university education ever since computer science and engineering entered the curriculum for undergraduate and postgraduate degrees. As the use of computer technology spread to all aspects of human activities,

E. Huebner (✉) and D. Bem
Computer Forensis Consulting, 324 GWH, Warrimoo, NSW 2774, Australia
e-mails: ehuebner@computerforensis.com; dbem@computerforensis.com

H. Cheung
University of Western Sydney, Locked Bag 1797, Penrith South DC NSW 1797, Australia
e-mail: h.cheung@uws.edu.au

E. Huebner and S. Zanero (eds.), *Open Source Software for Digital Forensics*,
DOI 10.1007/978-1-4419-5803-7_2,

various software packages also found application in teaching many unrelated disciplines. In the early days much of this software was created locally, mostly because of the non-standard hardware and operating systems available. Much has changed in this respect in the last 20-30 years as the hardware and operating systems market stabilised. For example in terms of operating systems, there is a virtual dual monopoly with the so called Wintel approach (Microsoft Windows running on Intel processor based computer) on one hand, and various distributions of Linux on the other. This duopoly is a good illustration of the dilemma faced by educators: should they use commercial or open source products, alone or in combination.

Academic communities always favoured free open source software. Cynically one might say that this is mostly because of perennial financial problems faced by most if not all universities. This is not entirely true as software manufacturers often offer heavily discounted or even free educational licences for their software to universities. This is not charity; students familiar with specific software products are likely to continue not only using them once they start their professional career, but also recommending them to others, increasing the market share for the producer. It is interesting to note that since 2006 Microsoft is offering the Windows Academic Program [10], which contains source programs for building the Windows kernel. This strategy has been obviously adopted to take advantage of this phenomenon.

It is our opinion that the preference for open source software in academia is based on the deeper desire for academic freedom and full control of the software tool afforded only by the open source philosophy. This attitude may have grown from early days when most of the software, including operating systems, had to be created in-house for the specific locally built hardware. With the growing complexity of software products it is no longer possible to maintain full local control over the software packages, and open source software offers the next best option with the possibility of local modifications, if necessary or desired. In comparison, a commercial product has to be used “as is” or not at all. There is of course the aspect of maintenance and continuity of the products, which is often seen as a drawback especially in the corporate world. This is much less of a problem in an academic environment, where people are much more prepared to assume a hands-on approach to software maintenance. The continuity is also not an issue, on the contrary academic programs are continually updated and new software tools are introduced routinely.

For students and academic teachers of computer science there is yet another incentive to use open source software. These students are future designers and implementers of software packages, and the ability to examine and manipulate the source of a substantial software package is an important part of their academic and professional development.

This aspect is even more pronounced for students of computer and network forensics. A professional forensic expert has to have complete confidence in the software used in obtaining evidence. This confidence may be based on peer acceptance of the product, whether open source or commercial. It is an important part of the university education for students to develop a questioning attitude and propensity

for critical inquiry. Only open source tools allow the students to examine and analyse the actual code. This has several benefits:

1. gaining deeper understanding of the results obtained with the use of the tools,
2. extending the knowledge of the structure of the media and systems under analysis,
3. and, last but not least, a better appreciation of the construction of a complex software product.

We saw all these benefits realised in the delivery of the advanced third year subject, Computer Forensics Workshop, which we discuss later as a case study illustrating the application of open source computer and network forensics tools in tertiary education.

2.2 Computer Forensics Software Tools

The increased incidence of computer crime and the growing realisation of its potential impact on critical activities of the society created a need for dedicated computer and network forensics software tools. Commercial software companies and the open software community responded to this need with a number of software products which continue to develop providing new functionality and more sophisticated tools. There are now several dominant commercial players in this market, for example Guidance Software [6], AccessData [1], ProDiscover [14], X-Ways [24] and many others.

There are well established forensic software tools, both open source and commercial, for example:

- The classic Coroner's Toolkit (TCT) [29] – a selection of programs by Farmer and Venema dating back to 1999.
- The Sleuth Kit and Autopsy Browser [27] – an updated and enhanced contemporary version of TCT.
- Linux Security Live CD distributions – BackTrack [3] INSERT Rescue Security Toolkit [7], Helix [28], DEFT [5], etc.
- EnCase® Forensic Modules by Guidance Software [6].
- ProDiscover® Forensics by Technology Pathways [14] (Basic Edition is freeware).
- FTK™ (Forensic Toolkit) by AccessData [1].
- X-Ways Forensics by X-Ways Software Technology AG [24].
- Paraben Forensics by Paraben Corporation [12].
- NTI Computer Incident Response Suite by Armor Forensics [11].

A detailed survey of the market is beyond the scope of this paper. However it is worth noticing that this relatively new market is far from being uniform. Most commercial companies release a family of products with often overlapping functionality, and similar sounding names. A typical example is Paraben

Corporation who lists on their home Web page over thirty computer forensics related software packages [12]. This commercial policy creates a certain effect often not appreciated by the general user. To use a tool for specific tasks a computer forensics examiner needs to be trained in one of the commercial courses offered by a vendor who was selected by a company the examiner works for. After receiving vendor training in Paraben software a computer forensics examiner becomes familiar with specific Paraben terminology and their range of tools. Such a person is reluctant to change to another set of tools, as they became a "Paraben guru" within their organisation. This situation is very unfortunate and only serves the marketing purpose of one specific vendor. A company which invested in training their examiners in one specific computer forensics package typically has very little incentive to look at other tools.

This commercial approach contrasts sharply with what open source software can offer. Using any set of open source tools does not prevent an examiner looking at other tools – on the contrary, each new or modified tool can be easily integrated with the existing environment. This is of course impossible with closed source tools from different vendors. The open source environment encourages the examiner to better understand what a specific tool does, how it does it, and are there other tools possibly better suited to the task at hand.

While the open source environment has distinctive advantages, realistically one needs to appreciate that it would require a bold and unpopular decision within a computer forensics investigative organisation to change from Windows to Linux environment. In our recent work [26] we proposed an innovative approach where two parallel environments are used in a virtual machine configuration: the host (or the main environment) is Linux, and a Windows system is installed as a virtual guest. In such an environment an acquired disk image which needs to be analysed is mounted on the host Linux, and can be accessed at the same time from two environments: Linux and Windows.

A unique advantage of such a setup is creating a mechanism for natural migration between Windows environment and tools to Linux platform without "burning bridges": every investigator working in the Linux host environment with Windows running as a guest would still have full access to the familiar Windows tools. Moreover the investigator is not forced to use any environment in preference to another. It is natural that an investigator working in a parallel Windows/Linux environment would start trying Linux tools, and benefit from the new and powerful utilities as well as techniques not available under Windows.

2.3 Case Study

To demonstrate the application of open source computer forensics software in tertiary education we will discuss the computer forensics specialisation for the Bachelor of Computer Science degree at the University of Western Sydney [18] which was designed in 2005, first offered in 2006, and delivered annually. Our motivation for introducing it was twofold. Firstly we could see the rapidly increasing demand for

computer forensics professionals, and secondly we wanted to reignite the interest of prospective students in computer science as it was in noticeable decline across the world after the Y2K bug and the dot-com crash.

We previously had a solid program in computer science with a specialisation in systems programming, so the groundwork on which to build a computer forensics stream was already there. Topics like operating systems internals, file systems, computer organisation, data representation, information security, computer networks and the operation of a computer system were adequately covered in existing subjects. For the computer forensics specialisation we introduced a new subject, Computer Forensics Workshop, which was designed from scratch to serve as a capstone for the stream.

To ensure that students obtained the maximum benefit from attending the workshop, we set prerequisite subjects, namely Operating Systems, Systems Administration Programming and Network Security. To obtain the specialization in computer forensics students also have to complete the following subjects: Computer Networks and Internets, Computer Security, Information Security, and a specialised subject dealing with law of evidence, delivered by the School of Law. This complements a generic computer science program, which covers all core topics recommended by the ACM/IEEE-CS Computer Science Curricula [2].

2.3.1 Computer Forensics Workshop - Content and Outcomes

Computer Forensics Workshop [19] is the capstone subject in the stream. It is delivered as a combination of weekly lectures and laboratory sessions. The hands-on component is obviously very important in a workshop based subject, so the laboratory sessions last for 4 hours, twice the time compared with other subjects. The assessment is mostly based on laboratory reports, which students complete in their own time. The reports are intended to reinforce the need for meticulous documentation of all investigative activities performed in the laboratory session. There is also an assignment, completed outside scheduled hours. Because of the practical nature of the material covered, we decided that a final written exam would not be a suitable assessment tool.

Unlike other fields in computer science, no guidelines or recommendations exist for computer forensics curricula, so we had to rely mostly on our own research and professional experience in the related fields. The same process was followed by other universities introducing computer forensics into their curricula at the time [30, 32, 35]. The characteristic feature of our approach was to focus on first principles instead of relying on specific forensic software tools. This was our motivation for adopting an open source software approach with the variety of forensic tools available, from suites of programs to single independent utilities.

The first topic we covered was media preparation and copying techniques, so that students understand the issues involved in the preparation of forensically clean storage media to accept image copies of suspect media as well as performing an image copy from multiple storage media types without altering the source media.

This topic allows for the application of manual techniques using the disk imaging dd utility [16] and industry forensic software suites.

The next topic was file system structures and file type identification techniques, analysis of time stamps as well as searching for and identifying hidden data. We chose to cover in detail the prevailing file system formats in Windows and Unix derivative systems, namely FAT, VFAT, NTFS, Ext2 and Ext3. Analysis of file systems is the mainstay of computer forensics, and we decided to expose students to all possible techniques, from direct examination of binary data to sophisticated software suites.

As a prerequisite to the Computer Forensics Workshop students complete three security related subjects: Computer Security, Information Security and Network Security. These subjects cover security issues exhaustively, so there was no need for a substantial security component. It was still important to impress on students the consequences of applying computer security measures, including cryptography and steganography, in a forensic setting. Again we attempted to expose students to a whole range of tools and techniques, and we were able to use many open source software tools for hiding and encrypting data, as well as analysis of such data.

We also introduced new computer forensics techniques like live system investigations and memory forensics. The former was also used to demonstrate how much information can be gained from the system without privileged access, and the latter to show that clear text including passwords can be extracted from a memory image. Naturally there are many tools for Unix derivative systems, including native system tools which are open source. There are free forensic tools for Windows [9], which are not open source.

Another important topic which has to be included in the curriculum is network forensics. This included, but was not limited to, analysis and reconstruction of network activity and web browsing, as well as extraction and reconstruction of emails. Again many open source software tools exist to support forensic investigation of networks, and we allowed students to select the tools themselves. We did it to encourage students to try new unfamiliar tools and to gain skills comparing various tools

The final topic was how to prepare a system and network to best support subsequent intrusion and activity detection. We wanted to make sure that students realise the importance of a forensic plan for any computer installation, and are able to formulate such a plan in various environments. This is different from securing and protecting the system, and deals with processes and procedures necessary for adequate incident response management.

2.3.2 Workshop Requirements

There are some technical issues in an educational environment which need to be resolved to provide a suitable laboratory environment for computer and networks forensics, as reported in [30]. Firstly, some of the investigative procedures require fully privileged access to the computer system. The best solution is to build a stand-alone dedicated laboratory with limited controlled access to the network to create a safe ‘sandbox’ environment. This was not generally possible in our case, as all laboratory

rooms serve many different purposes, and it would not be economically viable to limit the usage to one subject only. One exception was the network forensics, as we had access to a properly equipped and isolated laboratory dedicated to teaching networks.

We solved this problem by using Helix, an open source customised distribution of the Knoppix Live Linux CD [28], which boots from a CD ROM and uses memory-mapped disks. The content of the local hard disk is never changed, and it is easy to restore the normal environment for the next class by rebooting all systems. We also designed the laboratory work in such a way that in most cases privileged access to the system was not required.

There are also various freeware products, for example Sleuthkit and Autopsy [27], which are packaged in the Helix distribution. To enable students to use the laboratories after hours we also installed Sleuthkit and Autopsy on the local distribution of Linux, provided in all laboratory computers by default, giving students access to Sleuthkit commands. Students were also provided with relevant URLs to be able to download the products and install them on their home computers.

2.3.3 Laboratory Structure

The laboratory experience plays a central role in a workshop style subject. Students have to be able to test the knowledge they gained by performing practical tasks, which in a professional setting would be part of a computer forensics investigation. Our notes below reflect the content of Computer Forensics Workshop during previous years, as well as changes and modifications which we are including in the latest 2008 delivery.

In the course of a 13-week semester we provide ten laboratory sessions, each preceded by the relevant lecture presenting the theoretical background and providing students with sufficient knowledge to be able to handle the laboratory tasks. In addition the assignments bring together knowledge acquired in individual modules, and help the students to coalesce the experiences in the laboratory sessions into one coherent whole. The following laboratory sessions are scheduled:

1. FAT file system investigation,
2. EXTn file system investigation,
3. Media preparation and imaging,
4. NTFS file system investigation,
5. Network forensics (3 modules),
6. Applied cryptography (2 modules),
7. Live systems investigation.

2.3.3.1 FAT File System Investigation

For this laboratory session we decided to give students an opportunity to examine the image at the binary level using a hex editor. To make the task less onerous we selected the simplest of FAT systems, FAT12. Students were given an image of a

floppy diskette, and asked a number of questions regarding the structure of the volume represented by this image, including the creation of the human-readable representation of the FAT table. The answers were to be based strictly on a HEX/ASCII view of the image. The additional difficulty in interpretation of the binary data was the little-endian representation and the necessity to split some bytes into 4-bit nibbles for FAT12 structures. The students of computer science dealt with this task easily, but some of the externals needed time to recall the skills they haven't used for a while.

The product we chose was a well known hex editor - Tiny Hexer, a free tool provided by Markus Stephany from Germany. Unfortunately this tool is not developed any more and not available for download, however we have Markus' permission to use the last available version which is very stable and perfect for our purpose. In addition to viewing HEX representation, this tool comes with a number of macros, which help interpret the image assuming a number of file systems. Students could use the FAT12 macro to confirm their own findings. In the future we would look at other similar hex tools which could replace Tiny Hexer.

2.3.3.2 Ext File System Investigation

For the laboratory exercises we selected the simplest of the Ext file systems, i.e. Ext2. We based the student tasks on the excellent guide provided by Barry Grundy [31]. The guide provides a series of exercises based on a disk image with a number of partitions, including an Ext2 partition. There is also a log file provided, which contains output of various commands used while data was collected from the original disk. After determining the disk structure, students carve the image into separate partitions, and proceed to examine the Ext2 partition with a number of Sleuthkit tools, for example ffstat, fls and icat. Some native Linux commands are also used to examine tar archives and determine the types of files found in the image. As an extension of these activities, students repeat the investigation outside the scheduled lab using Autopsy.

2.3.3.3 NTFS File System Investigation

Again we based the student tasks on the same excellent guide provided by Barry Grundy [31], and we used the Linux environment exclusively to analyse the NTFS image. This session was the most difficult of the three dedicated to file systems, because NTFS is not only the most complex of the systems included, but the full NTFS documentation is not in public domain. It is thanks to organisations like the Linux NTFS Project [11] that more is known about the NTFS structure, and most of the features are documented [34] based on reverse-engineering. The analysis of the NTFS image is conducted using Sleuthkit commands. Students derive a double benefit from this session. Not only they gain appreciation of the NTFS file system, but also they see that it can be fully investigated using FLOSS tools.

2.3.3.4 Media Preparation and Imaging

The students were asked to acquire a disk image using two different approaches and different environments. The first exercise used the AIR (Automated Image and Restore) tool from the Helix live Linux CD. In the second exercise the workstation was booted to Windows environment and students used the "Capture & Add Image" option from ProDiscover Basic software package. The next more advanced exercise asked students to carve certain areas of data using ProDiscover Basic. In the Ext laboratory students performed data carving using a different image and the Sleuthkit open source tool kit. In this set of exercises the students saw that they can use close source or open source tools which are functionally equivalent.

2.3.3.5 Network Forensics Modules

In previous deliveries we were using a mixture of Ethernet connected machines separated from the university network by a Linux based router. The computers in the lab were organised into groups. All computers in a group were connected to a hub. The use of a hub enabled network traffic amongst the computers in a group to be captured. All the hubs were connected to the router via a switch. There were three laboratory sessions related to the three network forensics modules, respectively network activity reconstruction, web browsing activity reconstruction and email extraction and reconstruction

Network Activity Reconstruction

The lab tasks in this session required the students to create various types of network traffic on the lab's network. They included some network protocols, simple text files, document files, and image files. Students were asked to capture the network traffic and to perform initial reconstruction of the captured information to its original application format. An open source network monitoring tool, i.e. Wireshark [23], was used. Wireshark is an excellent tool in collecting data from a network and it has some facilities to help in reconstructing application data. For reconstruction of more complex captured data, a substantial amount of manual intervention may be required.

Web Browsing Activity Reconstruction

This session was on reconstructing web browsing activities from cached files in a user's computer, especially from cached files created by a web browser. Students were asked to create data to be collected and reconstructed, by performing a number of different browsing activities using the Internet Explorer web browser. The activities included browsing and searching, Internet shopping, downloading, and other popular Web browsing activities. Students were asked to select two of the browsing

activities to perform initial reconstructions, by using an open source or freeware tool. No particular tool was specified, and students chose and downloaded the tool of their preference. Two of tools mostly chosen by students were open source Pasco [13] and freeware Web Historian [21]. In addition, students were asked to reconstruct some of the Web browsing activities manually, without the help of a tool, so that they could appreciate the differences between the two approaches and could improve their understanding of the inner working of the tool.

Email Extraction and Reconstruction

This session consisted of two major tasks on extracting and reconstructing emails from email storage files, and capturing and reconstructing email network traffic. The email storage files used in this task were the Outlook Express DBX files. Students were asked to extract several emails from the files, using a hex editor and guided by some knowledge of the structures of the files. The DBX files were intentionally damaged so that they could not be opened by the original application. The second major task was to capture network traffic containing a user's Web based emails and to reconstruct the emails from the captured traffic. Similar to Network Activity Reconstructions, the open source tool Wireshark was used to capture the network traffic and to help in reconstructing the captured web based emails. Different Web based email software products have different ways in composing and sending the emails. Reconstructing the captured emails involved a substantial amount of manual manipulations and knowledge of the underlining interactions between the web browser and the remote email server.

The current lab arrangement is adequate in providing students with knowledge in network forensics mostly on a local area network environment. One of the shortcomings in only using physical machines is that it is difficult to expand the coverage to an internetworking environment. Thus we intend to shift to workstations running virtualisation. Initially, a more mature commercial visualisation software, VMware [20], will be used. An open source visualisation tool such as Xen [25] may also be considered in the future. This arrangement should allow for setting up a larger scale and more flexible virtual network system within the lab without the need to physically expanding to the real university maintained network. It is possible in such an environment to simulate a wide variety of network scenarios. The only trade off is that relatively powerful workstations and servers are required to support such an environment.

2.3.3.6 Applied Cryptography Modules

The example encryption system used in this session was the Microsoft Encrypted File System (EFS) bundled with the NTFS file system. The students follow the methodology of extracting EFS decrypted files from a live system presented in [33]. The method of extraction is built around a free software utility, Robocopy [8] which does not modify any metadata of the file system during extraction. To confirm that this approach is forensically sound, students have to obtain a hash value for the

captured data calculated before and after the extraction. This is the first time students are introduced to the concept of live system investigation, and they learn to appreciate that it is indispensable in obtaining complete information about the system being examined.

We are also currently preparing a series of exercises based on TrueCrypt [17], the open source disk encryption software for Windows Vista/XP, Mac OS X, and Linux. TrueCrypt is a very powerful tool which allows the creation of a virtual encrypted disk within a file, which in turn is mounted as a real disk. Another option which it offers is the encryption of the entire partition or storage device. TrueCrypt is particularly interesting from the computer forensics point of view as the TrueCrypt volume cannot be identified as hiding information because the data is very well randomised. We intend to make students aware that various commercial encryption breaking tools fail to recognise the TrueCrypt volumes as carrying encrypted data. What's more even if there is an attempt to decrypt the data, it is not possible to guess the key within a realistic time frame. In a practical exercise one group of students selects a very short encryption key (say 2 characters long), and another group selects a key with a more realistic length (over 6 characters long). A demo version of a commercial password breaking tool is then used to recover the keys with predictable results: the long one is impossible to retrieve within reasonable time.

2.3.3.7 Live System Analysis

The students were asked to collect as much data about a live system as possible in both the Windows and the Fedora Linux environments. This laboratory was not fully prescribed; the students were encouraged to explore options or commands not specifically mentioned in the instruction if they believed that such action would produce forensically interesting results.

The first part of the session used a set of free Windows tools created by Mark Russinovich and Bryce Cogswell, and available from Sysinternals Web [9]. To assure the soundness of the results the students were asked not to trust the cmd utility from the system being investigated, and to use the trusted command shell from the Helix Linux CD [28]. Sysinternals PsTools provide commands to list logged users, collect MAC times, list active processes, list executable files which opened the networks ports, and more. Additionally Helix provides the Windows Forensic Toolchest (WFT) [22] with more tools for collecting information about the system. Helix also offers the Investigator Notes module which allows storing date and time stamped notes on the external media, thus creating a proper forensically sound record of all activities.

In the second part of the session the workstations were booted in the Fedora Linux environment, and information was collected with the standard Linux commands. Thus the operating system and hardware specifications were collected with the command uname –a and uptime, MAC times were recorded with ls, active processes were listed with the ps aux command redirecting output to ps.txt file, and the powerful netstat command was used to determine open ports and associated

applications. This demonstrated to the students that Linux without any additional computer forensics tools offers a powerful set of commands capable of collecting information about the system.

2.4 Commercial Software Alternative

We have considered whether it is necessary to give students access to commercial computer forensics software packages. It seems obvious that being able to work with a commercial package would enhance the general student experience, and provide opportunities for various certifications. The real question is whether it is strictly necessary in order to give the students a well rounded education. After reviewing the market we decided to take a balanced approach with the predominance of free open source software. We believe that it is of greater benefit to a student to be able to work from 'first principles', and to be able to conduct the investigation using simple tools like hex editors and command line utilities. Some exposure to dedicated software tools is also desirable, but only after the students learn to operate at the lower level of abstraction. All commercial distributors offer specific training courses for their products, and it is the usual practice for employers to finance such courses for their professional employees. The market changes continuously and it is not the goal of university education to give students specific skills with specific software products. Rather the students should be given sufficient foundation knowledge and learning skills to be able to gain the full benefit from commercial training if and when it is appropriate.

The computer forensic software companies are mostly small organisations, and typically they are not willing to grant free educational licences. One notable exception is ProDiscover [4], which offers a cut-down version of their software free of charge. This version was included in some of the laboratory exercises, mainly to demonstrate basic data carving techniques.

Another popular commercial package is the Forensic Toolkit (FTK) from AccessData. This software can be installed and used on any machine in evaluation mode, which restricts the number of files which can be analysed. The current limit is 5000 files, which makes it unsuitable for commercial work, but does not affect the educational application, as the test cases studied are limited in size. We demonstrated FTK to students in a lecture, and we intended to use it more extensively for laboratory work in this year's delivery of the Computer Forensics Workshop. Unfortunately commercial companies frequently change their policies regarding educational or free use of commercial software. Here are some examples:

- EnCase from Guidance Software [6] distributes sets of EnCase software and sample files which are intended to be used together. No other files will open in the demo version of Encase, thus preventing unlicensed use. While this approach is reasonable it requires following in the footsteps of Guidance Software prepared training cases and thus is restrictive.

- ProDiscover Basic from Technology Pathways [14] offers unrestricted use of their "Basic" version, so exercises can be tailored to specific needs. The only small concern is that this software has not been updated for some time now and thus it is not clear what its future is.
- Forensic Toolkit AccessData offered a free version limited to 5000 files which was suitable for educational purposes. Unfortunately the new release (FTK 2.0) can not be downloaded from AccessData any more, and at this stage it is unclear whether there is any support for free restricted use.

Other commercial vendors simply refuse to provide any free versions of their software for educational purpose, typically ignoring any inquiries. This is easy to understand within a mindset of commercial vendors: all of them in addition to selling software also offer various very expensive training courses (typically around US$ 1,000 per day) and thus, most likely, their marketing department considers universities as undesirable competition.

2.5 Conclusions and Future Work

Based on the feedback we received from our students and our peers, the computer forensics stream we designed and implemented thoroughly prepares students to begin their professional career in computer forensics, starting as assistants for experienced investigators and developing into fully-fledged computer forensics professionals serving both industry and law enforcement agencies. The currently available computer forensics open source software fulfilled most of the needs we had for the practical part of the delivery. The students completing the subject gained thorough knowledge of the forensic process, and would have no problem applying this knowledge using whatever tools are available. The Computer Forensics Workshop was also well received by practising computing professionals who enrolled as so called non-award students. Based on our experience and materials we collected, a similar subject is currently being introduced at the Warsaw University of Technology.

The body of knowledge in computer forensics, similarly to computer science – the discipline it emerged from – grows at a very rapid pace. It means that the specific content of the workshop will have to be constantly reviewed and adjusted. Our growing research strength in computer forensics also contributes to this process. We envisage that in future deliveries we will have to dedicate more space to memory forensics, live systems investigations as well as storage and systems virtualisation. With the continuing development of the open source forensics software products we will be able to develop laboratory work to match these changes. We intend to have a close look at an alternative Sleuthkit interface, PTK [15] with the view to include it in the future deliveries. PTK is currently in beta stage and promises to implement numerous new features essential for forensics investigations which should offer a great deal of features like analysis, search and management of complex cases, etc.

Another area which we would like to expand is the law content of the computer forensics stream. In particular it would be beneficial to the students who are likely to serve as expert witnesses to have practice in presenting evidence in a court of law. This kind of practice is useful to students in all forensic disciplines; it is not specific to computer forensics.

To summarise, the Computer Forensics Workshop was a demonstrable success. It is clear that it fulfilled students' expectations, and provided them with skills and knowledge that they will be able to apply in their professional life. This overall experience was enriched by the laboratory work based predominantly on FLOSS software products.

References

1. AccessData (2008). http://www.accessdata.com/. Accessed 2 March 2008
2. ACM Computing Curricula 2001 Computer Science (2001). http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf. Accessed 3 December 2006
3. BackTrack, Remote-Exploit.org. http://www.remote-exploit.org/backtrack.html. Accessed 1 August 2007
4. Computer Forensic Tool for Law Enforcement (2006). http://www.techpathways.com/ProDiscoverDFT.htm. Accessed 20 October 2006
5. DEFT (2008) http://deft.yourside.it/index.php. Accessed 18 February 2008
6. EnCase Forensic Modules (2007) http://www.guidancesoftware.com/products/ef_modules.asp. Accessed 17 December 2007
7. INSERT Inside Security Rescue Toolkit. http://www.inside-security.de/insert_en.html. Accessed 28 November 2007
8. Microsoft, Alternatives to the Directory Replicator Service (2006)
9. Microsoft, Windows Sysinternals (2007) http://www.microsoft.com/technet/sysinternals/default.mspx. Accessed 12 June 2007
10. Microsoft, Windows Research Kernel (2006) www.microsoft.com/WindowsAcademic. Accessed 15 November 2008
11. NTI (2008) http://www.forensics-intl.com/index.html. Accessed on 30 June 2007
12. Paraben Corporation (2008) http://www.paraben.com/. Accessed on 30 March 2008
13. Pasco(2008)http://sourceforge.net/project/shownotes.php?release_id=152387&group_id=78332. Accessed 21 June 2008
14. ProDiscover Forensics (2006) http://www.techpathways.com/ProDiscoverDFT.htm. Accessed 20 October 2006
15. PTK an alternative Sleuthkit interface - DFLabs (2008) http://ptk.dflabs.com/. Accessed 31 August 2008
16. The Open Group Base Specifications Issue 6 (2004) http://www.opengroup.org/online-pubs/009695399/utilities/dd.html. Accessed 21 March 2007
17. True Crypt - Free Open-Source On-The-Fly Disk Encryption Software (2007) http://www.truecrypt.org/. Accessed 15 January 2007
18. University of Western Sydney Handbook (2008) http://handbook.uws.edu.au/hbook/course.asp?course=3506. Accessed 6 February 2008
19. University of Western Sydney Handbook – units (2008) http://handbook.uws.edu.au/hbook/unit.asp?unit=300447.1. Accessed 10 March 2008
20. VMware. http://www.vmware.com/. Accessed 10 March 2008
21. Web Historian http://www.download.com/Web-Historian/3000-2653_4-10373157.html?part=dl-RedCliffW&subj=dl&tag=button&cdlPid=10562519. Accessed 21 June 2008

22. Windows Forensic Toolchest (WFT) (2007) http://www.foolmoon.net/security/wft/. Accessed on 15 June 2008
23. Wireshark (2008) http://www.wireshark.org/about.html. Accessed 20 November 2007
24. X-Ways Software Technology AG (2007) http://www.winhex.com/. Accessed 7 October 2006
25. Xen. http://xen.org/. Accessed 9 June 2007
26. Bem D (2008) Open Source Virtual Environments in Computer Forensics, the 1st Workshop on Open Source Software for Computer and Network Forensics, Milan , Italy
27. Carrier B (2007) The Sleuth Kit. http://www.sleuthkit.org/sleuthkit/desc.php. Accessed 10 February 2007
28. E-fence, The HELIX Live CD (2007) http://www.e-fense.com/helix/. Accessed 9 Feb. 2007
29. Farmer D, Venema W (2007) The Coroner's Toolkit (TCT). http://www.porcupine.org/forensics/tct.html. Accessed 25 April 2007
30. Gottschalk L, Liu J, Dathan B, Fitzgerald S, Stein M (2005) Computer forensics programs in higher education: a preliminary study, SIGCSE Technical Symposium on Computer Science Education, 203–231
31. Grundy BJ (2007) The Law Enforcement and Forensic Examiner – Introduction to Linux – A Beginner's Guide to Linux as a Forensic Platform. http://www.linuxleo.com/Docs/linuxintro-LEFE-3.21.pdf. Accessed on 17 March 2008
32. Hentea M, Dhillon HS, Dhillon M (2006) Towards Changes in Information Security Education. Journal of Information Technology Education 5:221–233
33. Huebner E and Bem D (2008) Forensic Extraction of EFS Encrypted Files in Live System Investigation, Journal of Digital Forensic Practice 2:1–12
34. Russon R, Fledel Y (2004) NTFS Documentation, Free Software Foundation, Inc.
35. Yasinsac A, Erbacher RF, Marks DG, Pollitt MM, Sommer PM (2003), Computer Forensics Education. IEEE Security and Privacy 1(4):15–23

http://www.springer.com/978-1-4419-5802-0

Open Source Software for Digital Forensics
Huebner, E.; Zanero, S. (Eds.)
2010, VII, 124 p., Hardcover
ISBN: 978-1-4419-5802-0