Global Journal of Information Technology: Emerging Technologies

Volume 11, Issue 2, (2021) 29-35

www.gjit.eu

Computer forensics teaching for undergraduate students

Wenjuan Xu*, Imperial College of London, Graduate Tunnel Engineer, London, UK

Suggested Citation:
Xu, W. (2021). Computer forensics teaching for undergraduate students. *Global Journal of Information Technology: Emerging Technologies. 11*(2), 29–35. https://doi.org/10.18844/gjit.v11i2.5266

Received from July 23, 2021; revised from August 02, 2021; accepted from October 25, 2021.
Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, Cyprus.

Abstract

Computer forensics is not only important for the security professionals in the forensics field, but it can also help the information technology professionals working in the network administration, system management and other related fields. Our department has a computer forensics course offered to undergraduate students to prepare them to be successful in their future career. In this paper, we introduce our forensics course design, especially we explain our course's hands-on activities, which include the labs and the projects for practicing the computer forensics-related programming and the existing tools using. In this paper, we have already introduced our secure computing and information assurance 470 course in detail. We explain how we design our programming labs and existing computer forensics tools' labs. Also, we describe the two project assignments based on these labs. Through these labs and project design, students have the feedback such as gaining better understanding about the computer forensics, building their programming skills and improving their hands-on ability.

Keywords: Forensics, experiential learning, Python, tool, evidence.

* ADDRESS FOR CORRESPONDENCE: **Wenjuan Xu**, Imperial College of London, Graduate Tunnel Engineer, London, United Kingdom.
E-mail address: wxu@frostburg.edu

1. Introduction

Due to the need for more information security professionals in the IT department, system design and development department, testing and other computer-related fields, our department developed a new undergraduate degree called secure computing and information assurance (SCIA) in 2014. The main purpose of this programme is to educate our undergraduate students to be security professionals, who can be exposed to the different knowledge and tools in the security field and can pursue a successful career and take responsibility for the overall health and security of the cyber world. In our degree's design, we cover software security, database security, network security, computer and network forensics and other security aspects. In this paper, we will focus on sharing our designing and teaching experience of the forensics course. Many universities offer the computer forensics courses. Our SCIA degree has two computer and network forensics courses, including Computer and Network Forensics I – SCIA 470 and Computer and Network Forensics II – SCIA 471, which are all required courses for students majoring in SCIA. SCIA 471 focuses on network forensics, which is introduced in our previous work [8] about our teaching method. On the other hand, SCIA 470 focuses on computer forensics, including collecting and analysing the different evidence on the computer hard drive, computer system, mobile device, other electronic device, cloud, social media and so on. In this paper, we will mainly explain how we design and teach the SCIA 470 course.

In the forensics field, hands-on experiential learning is critical for students to understand the fundamentals better. Especially, students can learn how to operate in the forensics environment and handle the forensics cases in the real world. Hence, we design this course as an experimental leaded learning orientation. Other than the lecture, we generally design this course with two aspects, including programming and the existing computer forensics tools using. For each part, we have different labs and a corresponding project. To facilitate the work, we apply virtualisation technologies [9] to our teaching. With the existing virtualisation technology, we can easily have different operating systems with different hard drives. In addition, we can obtain better host security, work status saving and easy hardware management. In our course, we mainly use the VMware desktop software for running multiple instances of X86 or X86-64 compatible operating systems on a single physical computer. In this paper, we first introduce our course curriculum design. Next, we explain the details about programming-related experiential learning in the computer forensics including the various labs and the corresponding project. Then, the lab design about different forensics tools and the related project will be elaborated. Finally, we summarise our course design and the future work.

2. Curriculum introduction

SCIA 470 Computer and Network Forensics I is a three-credit course offered in the fall of every year. The course catalogue is described as 'Forensics tools, methods and procedures used for investigation of computers, techniques of data recovery and evidence collection, protection of evidence, expert witness skills and computer crime investigation techniques. Analysis of various file systems and specialised diagnostic software used to retrieve data'. The prerequisite of this course is SCIA 210 Introduction to cyber law and SCIA 360 Operating System Security. Upon completion of the course, the students can gain these six main goals shown in Table 1.

Table 1.

Describe the basic concepts of computer forensics.	Know the main process of computer forensics and explain the legal constraints in the process.
Demonstrate the ways of gathering computer evidence and other related digital evidence.	Use basic programming language to work on the forensics evidence collection and analysis.
Understand the forensics challenges facing with more and more evidence from different new fields such as cloud and social media.	Use different forensics tools to collect the forensics evidence and analyse that.

In trying to accomplish these course objectives, we adopt the textbook – Guide to Computer Forensics and Investigations (Nelson, Phillips & Steuart, 2018). In addition, we supplement materials about computer forensics programming. In total, we have the topics covered as shown in Table 2. In the 14-week semester course, we cover these 15 topics, and correspondingly to each topic, we have homework and exams for helping the students understand computer forensics fundamentals and methodologies. In computer forensics, to help the students have hands-on abilities, we design two main parts of the lab and the project. The first part is programming and the second part is tool using. For some weeks, we assign the labs to learn about how to use the existing computer forensics tools and on the other weeks, we ask the students to practice how to programme for computer forensics. Based on these labs, we then have two projects including programming and using existing forensics tools for analysing a given forensics scenario.

Table 2

Understanding forensics profession and investigations	The investigator's office and laboratory	Data acquisition	Processing crime and incident scenes
Current forensics tools	Working with Windows and CLI systems	Linux and Macintosh file systems	Recovering graphics files
Forensics analysis and validation	Live acquisitions and network forensics	E-mail and social media investigations	Mobile device forensics
Cloud forensics	Forensics programming	Report writing for high-tech investigations	

3. Programming for computer forensics

There are different programming languages in the security fields such as Java and Python. In our course, we use Python as the main programming language due to its unique features such that it has simplicity of syntax, comprehensive inbuilt modules, strong help and excellent support from the developer's and user's community.

3.1. Programming labs

In our course, we build our python labs supported by the different Python libraries covering the following aspects:

- *Hashing.* Python has a hash library supported for generating the hash value for the evidence. The students can call the functions from the library, generate the hash of the evidence and then compare the hash value to check the integrity of the evidence.
- *Indexing and searching*. Python indexing is to help take a complete look at the evidence and gather potential evidence. Following the indexing lab, we have the keyword searching lab, which helps to analyse the forensics evidence based on the searching results.
- *Cracking an encryption.* Files can be encrypted with different algorithms and need to be decrypted in the forensics analysis. Python has a library Python cryptography toolkit, which is a collection of both secure functions and various encryption algorithms such as AES, DES, RSA, Eltamal etc. We can then easily use this library to perform encryption with the python scripts.
- *Carving an image file*. Python imaging library adds the image processing ability to Python. Students download this library into their python and then can perform forensics image processing and carving. For example, they have a lab to recover the impaired image and collect fragments of an image.

- *Mobile forensics*. When a mobile phone is encrypted, with the given hash table generated based on the possible passwords, students practice how to write python code to crack the screen-locking password to retrieve data from a smartphone using python.
- *Working with Python and Scapy*. Although the course focus is computer forensics, we also give a brief overview about network forensics, in which one of the important tasks is analysing the network data packets. With the Python Scapy library [19], we can analyse the network data packet to identify if an attacker is performing SQL injection [20].

3.2. Programming project

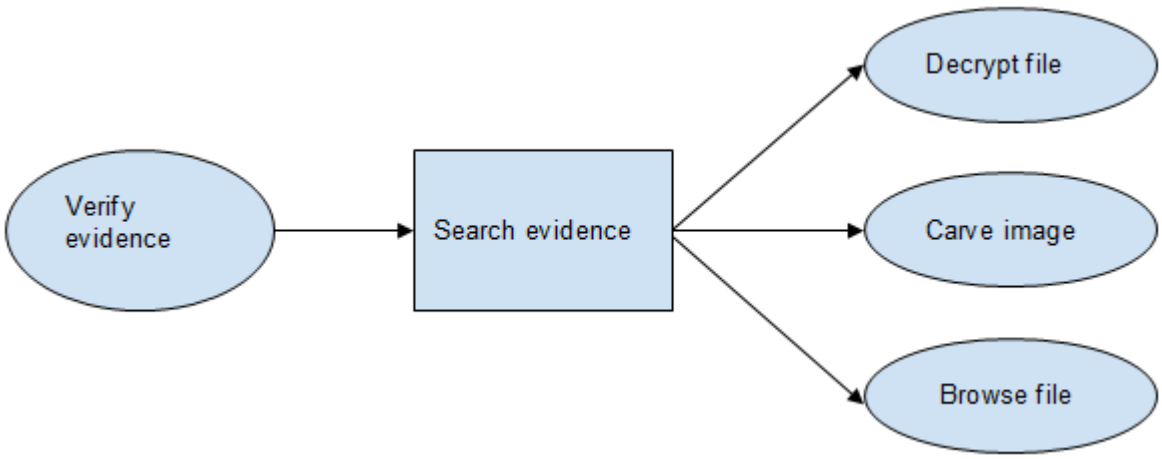

Figure 1.

To help students summarise what they have learned in the programming labs, we challenge the students with a given scenario, in which we provide different evidence in the virtual disk. As shown in Figure 1, students are required to use python to generate a copy of the evidence, verifying the evidence, and then searching the evidence with keywords. Then based on the searching results, students need to crack the encrypted files, carve the impaired image file and browser through other searching results. Due to the complexity of the programming, we can also see that there are a lot of limitations if only using programming in the forensics analysis. In the following section, we explain how we design the labs and projects with existing computer forensics tools.

4. Computer forensics tools

The main purpose of computer forensics is to search, preserve and analyse the information on the computer systems to find the potential evidence for the different cases. With computers getting more and more powerful, computer forensics tools are evolving rapidly and there are a lot of different forensics tools used to perform computer forensics, covering all kinds of aspects. In our class, we adopt the current popular and powerful forensics tools running on Windows, including Access Data [21], Encase [22] and Prodiscover [23], and the Linux forensics tool, such as Sluetkit [24]. In addition, we also use some auxiliary tools such as Hex Editor [25] and hash calculation related tools for forensics analysis.

4.1. Forensics lab design

Computer evidence is mainly stored on the different devices such as hard drive, cloud drive, USB or camera and so on. We designed our first lab to help students understand how the data is stored. Then

we have our forensics labs with the forensics work flows, including capturing, analysing and finally generating a report.

1. *Understanding the data store.* This can help the students learn how to build a clean hard drive, how to look at the details in the drive, and how to repair a hard drive. In this related lab topic, students are required to use a virtual hard disk, Hex editor and other related files to work on the hard drive and observe the data changing in the drive.
2. *Capturing the evidence.* There are different forensics tools used in this lab topic. On the top of the windows platform, we separately use Encase, Access Data (FTK imager) and Prodiscover for full or partial capturing evidence from the computer hard drive, register, USB and mobile device. We also require the students understanding the use of the hash algorithms for evidence capturing validation. On top of the Kali, we used Sleukit for verifying and capturing the evidence.
3. *Analysing the evidence*. This plays a critical part in our lab design. Other than the searching function, we need to perform data decryption, file carving, image file carving and email decoding. On Windows, we used FTK analyser, FTK register, FTK decryption, Encase and Prodiscover to perform the related works separately. As the continuous lab of capturing the evidence, in Kali, we continue to use Sleukit to analyse the captured image.
4. *Generating a report*. Prodiscover, Encase and Sleukit all can generate a report automatically after the analysis. Students need to further explain the report.

4.2. Semester project sample

As shown in Figure 2, in the semester project, we distribute a forensic virtual disk file, in which there are encrypted files, unknown files, impaired files, email, mobile data, cloud data, social media data and other information. Students are required to use at least two forensics tools for capturing and analysing the evidence, comparing the results, and generating a report. Based on the students' feedback, their knowledge and ability are improved through working on the project with reviewing the labs. Also, comparing Figures 1 and 2, we can see that with the existing forensics tools, we can have a more comprehensive forensics analysis, especially with the existing tools, we can easily analyse different types of files and generate a forensics report. On the other side, we can also see that with the emergence of more and more complexity of the forensics files, the analysis will be more complex and we need more powerful forensics tools to work on.

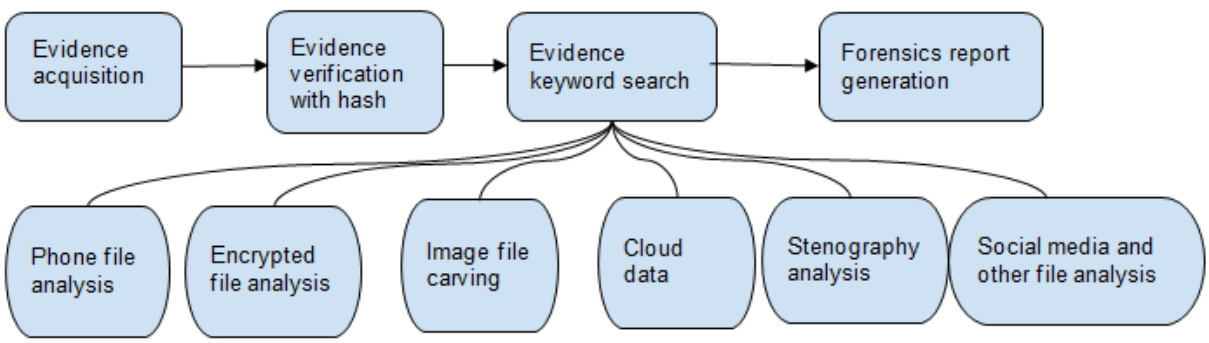

Figure 2.

5. Conclusion and future work

In this paper, we have already introduced our SCIA 470 course with details. We explain how to design our programming labs and existing computer forensics tools labs. Also, we describe the two project assignments based on these labs. Through these labs and project design, students have the feedback such as gaining better understanding about the computer forensics, building their programming skills and improving their hands-on ability and so on. In the near future, we plan to have

a computer forensics platform for the students to practice tasks with programming and existing tools working together. Also, we are going to evaluate the different forensics tools used in the teaching, check the student's acceptability of these tools and further improve our lab and project design.

References

Access Data. Retrieved from https://accessdata.com/

Armstrong, C. & Jayaratna, N. (2004, June). *Teaching computer forensics, uniting practice with intellect.* Proceedings of the 8th Colloquium for Information Systems Security Education. West Point, New York.

Association of Chief Police Officers (ACPO). (2003).*Good practice guide for computer based electronic evidence.* London, UK: NHTCU.

Bem, D. & Huebner, E. (2008).*Computer forensics workshop for undergraduate students* (vol. 78). Proceedings of the Tenth Conference on Australasian Computing Education. Australia: Australian Computer Society, Inc.

DES, AES PY. Retrieved from https://www.dlitz.net/software/pycrypto/api/current/Crypto.Cipher.DES-module.html

Encase. Retrieved from https://www.guidancesoftware.com/encase-forensic

Hex Editor. Retrieved from https://mh-nexus.de/en/hxd/

Java. Retrieved from https://www.java.com/en/

Kessler, G. C. & Shirling, M. E. (2006). The design of an undergraduate degree program in computer and digital forensics. *The Journal of Digital Forensics, Security and Laws, 1*(3), 3.

Lim, N. (2006). Crime investigation: a course in computer forensics. *Communications of the Association for Information Systems, 18*, 10.

Lunsford, D. L. (2009). Virtualization technologies in information systems education. *Journal of Information Systems Education, 20*(3), 339.

Nelson, B., Phillips, A. & Steuart, C. (2018). *Guide to computer forensics and investigations* (6th ed.).

Prodiscover. Retrieved from https://www.prodiscover.com/

Pycrypto. Retrieved from https://pypi.org/project/pycrypto/

Python. Retrieved from https://www.python.org/

Python Imaging Library. Retrieved from http://www.pythonware.com/products/pil/

Python Scapy. Retrieved from https://pypi.org/project/scapy/

Sleukit. Retrieved from http://www.sleuthkit.org/

Snyder, R. M. (2007, September). *Security programming using Python: man-in-the-middle attacks* (pp. 1–6). Proceedings of the 4th Annual Conference on Information Security Curriculum Development. New York, NY: Association for Computing Machinery.

SQL injection.

Srinivasan, S. (2013). Digital forensics curriculum in security education. *Journal of Information Technology Education, 12*, 147–157.

Virtual Box. Retrieved from virtualbox.com

VMware. Retrieved from vmware.com.

Xu, W. & Pan, X. (2017). *Teach network forensics course with experiential learning.* Proceedings of the 11th International Multi-Conference on Society, Cybernetics and Informatics: IMSCI 2017.

Yasinsac, A., Erbacher, R. F., Marks, D. G., Pollitt, M. M. & Sommer, P. M. (2003). Computer forensics education. *IEEE Security and Privacy, 1*(4), 15–23.