Ethical Dilemmas in Teaching Computer and Internet Security

Brian Tompsett

Department of Computer Science, University of Hull
b.c.tompsett@hull.ac.uk

Abstract. This paper could be subtitled "Are we teaching the next generation of computer criminals and internet terrorists"? This issue was raised by the Security Services as part of the collaborative network meeting in the area of IT Forensics and Data Analysis hosted by City University. These are valid concerns about the nature of material taught to computer science students in the area of security.

The questions are also important ethical dilemmas for any professional working in the computer and internet security field. These are also applicable when discussing such security risks with the media, members of the public and even legislators. Information on vulnerabilities has to be presented so that it informs programmers and computer users about the areas of risk, but without providing recipes for them to use to conduct criminal activities or mischief themselves.

The paper will look at several case studies from the curriculum at the University of Hull at both undergraduate and postgraduate level. Some specific problem areas of email forgery, security of the Windows operating system and exploitation of buffer overflows, and deception in online auctions, will be explored.

Keywords: Security, Teaching, Ethics.

1 Introduction

Students at Hull perform an assessed practical where they evaluate the security of existing laboratory computers, which allows them to put the security theory into a practical context. The response of system managers to this exercise is interesting; although they cannot influence the taught component of a course, they express their discomfort with the practical aspects. Despite this, the students find the practical aspect valuable, and each year a new set of vulnerabilities in our laboratory computers in exposed. This enables the configuration of laboratory computers to be improved on an ongoing basis. The experience does show that organisations often prefer to believe that their computers are secure, and find any information to the contrary painful. Security by obscurity seems to be the desire of management within organisations, which in the security business in known to be no security at all.

It is an interesting question, "what do computer science students know about computer security and vulnerabilities?" Discussions in the media resulting from recent

D. Weerasinghe (Ed.): ISDF 2009, LNICST 41, pp. 76–81, 2010.

computer hacking and attacks [1], [2] have shown that there is a general view that those attacking computers have a special form of arcane knowledge which is not taught. In fact, this material is taught; since we need to improve the knowledge base of all computer graduates to enable them to construct better and more reliable systems. Such systems are crucial since computer users, on the other hand, are not supplied with knowledge that would enable them to protect themselves from being compromised. Furthermore, the knowledge necessary to protect against such attacks on existing systems is rarely included in a standard computer curriculum, or anywhere else in a standard educational programme. In short, we tend to tell computer science students much about building new systems but little about protecting existing ones, which we tell computer users virtually nothing about either.

This paper results from meetings and discussions which were part of the EPSRC research network in IT Forensics and Data Analysis hosted by City University in London from 2008 to 2009. The research network is composed of academics from around the UK research community and also involves the security services. Meetings of the research network examined research topics of mutual interest which included visual or audio analysis, data mining, forensic analysis, computer and network security and so on. Although the group was constituted primarily to facilitate research networking, the topic that appeared to be of most concern to the security services was not research into new criminal detection techniques, but a concern over the curriculum taught to computer science students.

The paper does not attempt to thoroughly explore this wide and contentious topic, but rather to highlight certain issues by way of a quick tour of the subject and to focus on a couple of specific areas by way of example from the topics in programming, e-business, networking, and security.

2 The Issues in Computer Teaching

Software development is a key part of the computer science where the issues of vulnerabilities due to software flaws arise. Students need to be aware of these from the programming and engineering perspectives in that students need to be taught what are common mistakes to make and which ones have been made in the past. The mechanism of the flaws needs to be explained so the students can understand why it is an issue, and also how it can be solved.

In initial teaching of computing and software engineering, the focus is more on the correct implementation of a specification, the crafting of data structures, algorithms, the design user interfaces and databases. These subjects do not expose many ethical dilemmas where students could be taught inappropriate material.

When teaching internet and distributed computing however, the ethical issues begin to appear. The interconnection of computers with the internet exposes the smallest flaws in computer software to a severe test, since each application is itself another vulnerable portal to the computer system as whole. Every networked application or interactive web page made by a student could potentially compromise a whole computer system. This is much more of a security vulnerability than the problems of software development in unconnected systems.

When the topic becomes the operation and implementation of network protocols and internet applications the ethical questions become more acute, and it is these topics that the paper will examine by way of some examples from current curricula.

3 Teaching Network Programming

The basic tenets of good practice and software engineering are taught to students at the beginning of most computer courses in the selected initial teaching language and, later, often in their second year, they learn to build applications that use the network, by initially programming a simple client and server application. The next stage is to develop an understanding of multithreading and concurrency and the associated issues of deadlock, as well as connection timeouts, in order to make a robust and functional server. They will also learn, as a side effect, about resource starvation and denial of service attacks. These become evident when a student is shown how to properly test their client and server implementations for multiple connections and responsiveness. Along with this they also need to understand the issues regarding interoperability and portability of solutions, since a server that only talks to a single client and *vice versa* is not a realistic solution.

This might seem to be uncontentious in terms of the ethical issues it raises, but can lead to knowledge of how to attack systems since knowledge about constructing a robust system often leads to an understanding of the vulnerabilities. It is not far from the teaching laboratory experience to discover that a denial of service attack or resource starvation is not detected or prevented by the computer system or network infrastructure and that, further, they can easily either by accident or deliberately attack servers run for the class, those operated by other students, those run operationally by the university (such as the main web server) and any server of any kind elsewhere on the network. We can only explain that it is wrong to do so!

4 Teaching e-Business

Many computer science (and business) courses include material on e-Business. In teaching the material issues of trust need to be covered, particularly when explaining online identities and money transfer systems. The ease with which fraud can be conducted online becomes apparent to the students, and may become a temptation to them. Further, when teaching aspects of online marketing, the ability to reach a large number of people is also evident, and the students may also see an attraction in undesirable technical-based marketing, such as spam and ad-ware.

5 Teaching Internet Protocols

Network protocols are considered in a layered manner using the OSI 7 layer notation. In the following subsections a few sample problem areas in the teaching of each layer are examined.

5.1 Datalink Layer

The main elements at the datalink layer that students become familiar with in a basic computer networking course on internet protocols are the dynamic address assignment protocols such as ARP, RARP, dhcp and perhaps bootp. This does expose the students to the notion that they could implement servers to answer these requests or monitor this traffic and the information it contains.

5.2 Network Layer

The IP layer is fairly straightforward and its teaching does not expose too many ethical issues, but to understand the operation of IP the rôle of ICMP needs to be explained. The use of the TTL field, the ICMP echo and error responses and their application in programs such as Ping and Traceroute are essential for a student to understand. Understanding of these mechanisms can allow a student to probe network topologies or conduct denial of service attacks [3]. The students may also discover that most computer systems and network infrastructure will not interfere with these activities, however, coverage of these topics is essential to a proper computer networking course. When considering IP addressing students may also be exposed to the concepts of IP address spoofing, as it becomes clear when the mechanisms of routing are covered that the relationship between datalink address and IP address is not fixed.

5.3 Transport Layer

TCP contains several features that, when explained, expose security issues with the protocol [4]. To understand the operation of the protocol students need to understand the acknowledgement system, which includes the numbering of the bytes and the reasons for not starting the numbering at a known point. The connection orientation and the open and close protocols also expose mechanisms of resource starvation and denial of service attacks, particularly when combined with an explanation of the state diagram of the protocol. The state change mechanism for the protocol is flawed and this becomes obvious as that aspect of the protocol is taught.

5.4 Application Layer

It is essential in a networking course to explain the operation of basic internet application protocols. Those often taught include telnet, ftp, smtp (for email) and http (for the web). In explaining the operation of the protocol for technically oriented computing students some exploitable flaws in these applications will become self-evident. For example, when covering telnet and ftp the unencrypted nature of user names and passwords is exposed. The http protocol is fairly straight forward, and its use as the backbone of the web means that many potential weak areas have all been closed and made unavailable to experimenting students; it therefore does not present so many ethical issues in teaching. Smtp, conversely, is full of flaws that stand out and whose symptoms are evident to any user of email. When covering the protocol it is necessary to explain that the sender and the host details are not authenticated and can be forged, and that the entire email headers and body can be user-supplied. Students can then see how spam can easily be generated, but by implication they can also see how easy it is to send. It also is clearly visible to the student how denial of service attacks using email (mail bombs) can be generated.

6 Teaching Trustworthiness and Security

At higher levels in computer science teaching, usually at the master's level, students need to know how to build systems that are trustworthy and secure. At the undergraduate level the students will have a full education in software engineering and computer networking protocols this is not sufficient for them to understand how to engineer fully secure and trustworthy systems. Students need to understand mechanisms of cryptographic security including public key encryption. They then look at the use of encryption in networking such as the secure socket layer (SSL) and possible flaws in key exchange. This exposes students to possible weaknesses in standard encryption and key exchange systems in common usage. Similar weaknesses in encryption in wireless protocols make students aware of mechanisms to attack WEP and WPA.

In software trustworthiness students need to be aware of the rôle of buffer overflow in system compromise, as well as how attack vectors for buffer overflows are constructed and exploited so they can protect against them. Other prominent flaws covered are ones that can be found commonly in operating systems such as timing race conditions, and poor input validation in internet tools. One often overlooked area is the mathematics of randomness that affects the security of operations such as online gambling. Students also consider fully trustworthy systems mechanisms such as a TPM chip [5] which provide onboard encryption and key escrow to authenticate booting software chains. Knowledge of these areas allows students to have the potential to fully exploit all the weaknesses of computer systems, but also allows them to construct systems that are resistant to such vulnerabilities, which is the pedagogic goal of the material.

7 Conclusions

There has been recent discussion of these matters in the professional press [6], [7] which shows that the dilemmas are still omnipresent. When the House of Lords Science and Technology Committee considered Personal Internet Security [8], [9] they also exposed aspects of the problem.

The law enforcement and security community seems to be unaware that knowledge of computer insecurity is widespread and the frequently used with inappropriately. As suggested by the House of Lords, the environment for reporting, detecting and prosecuting these offences needs improvement in order to make the chances being caught greater.

It is clear that it is a necessity to retain within our curricula coverage of technical flaws within computer systems and to educate future developers on how to avoid them. I would go further and say that we should also explain how to protect users and developers from flaws in existing systems. Above all, we need to educate students on their ethical responsibilities in how they use their knowledge.

References

1. Boyd, C.: Profile: Gary McKinnon, BBC News Online (30 July 2008), http://news.bbc.co.uk/1/hi/technology/4715612.stm
2. BBC News: Estonia fines man for ‘cyber war’, BBC News Online (25 January 2008), http://news.bbc.co.uk/1/hi/technology/7208511.stm
3. Gont, F.: Security Assessment of the Internet Protocol, CPNI (July 2008), http://www.cpni.gov.uk/Docs/InternetProtocol.pdf
4. CPNI: Security Assessment of the Transmission Control Protocol, CPNI Technical Note 3/2009 (2009), http://www.cpni.gov.uk/Docs/tn-03-09-security-assessment-TCP.pdf
5. Trusted Computing Group: Trusted Platform Module (2009), http://www.trustedcomputinggroup.org/developers/trusted_platform_module
6. Li, H.K.: Security Ethics, Letter to the Editor, IT Now, British Computer Society, p. 32 (May 2009)
7. Wanigaratne, S.: Teaching Security, Letter to the Editor, IT Now, British Computer Society, p. 32 (July 2009)
8. House of Lords Science and Technology Committee: Personal Internet Security, 5th Report of Session 2006-07, HL Paper 165, TSO (2007)
9. House of Lords Science and Technology Committee: Personal Internet Security: Follow-up, 4th Report of Session 2007-2008, HL Paper 131, TSO (2008)