Practical Assignments in IT Security for Contemporary Higher Education

An Experiment in Exploiting Student Initiative

Alan Davidson and Kjell Näckros

Department of Computer and Systems Sciences,
Stockholm University/Royal Institute of Technology, Sweden
{alan,kjellna}@dsv.su.se

Abstract. Modern university studies cater to large groups of students with consideable variation in background knowledge. This creates problems when designing viable practical exercises, not least for the subject of IT Security. We address these problems by creating a study environment within which students have the freedom to design and execute their own exercises. We suggest and test ideas for providing sufficient motivation and structure for student activity while minimising the need and cost for staff intervention.

Key words: IT Security, Practical Assignments, Pedagogics, Constructionist Learning

1 Introduction

Practical projects are of considerable pedagogical value for any learning environment, not least for university courses on IT Security. At the Department of Computer and Systems Sciences it has been our ambition to keep a practical element to our IT Security introductory courses, such as on the first, five week subject overview course of the Master's Programme in Information Security. During such courses we would like to tie several the more theoretical themes into a practical exercise. Since there is a lot of ground to cover in an overview course, and since most practical exercises concentrate on relatively narrow aspects of the subject, we do not believe it is motivated to spend more than one week's study time out of the five on practicals.

Some examples that were considered as candidates included red-team/blue team exercises and capture the flag competitions such as presented in [7], or the coding of buffer overflow exploits such as in [6]. There are however two main problems with this ambition:

1. Students who come to our courses have very mixed backgrounds. Though we have previously had many further educational students from industry, of late our subject matter has become far more popular with academic students. Neither of these groups can be expected to necessarily have, for example,

any knowledge of network protocols, computer architectures, programming, or any other operating system than Windows. We also have increasing numbers of international students, implying increased language difficulties and diversity in cultural background. All these factors combine and ultimatley make it very difficult to find any specific exercises that can stimulate all students - not without having to provide extensive background information and continuous assistance.

2. Practical experiments in IT Security are very sensitive to the computing environments used, and in the real world, these environments change rapidly. For example, for the buffer overflow exploit exercise mentioned above, after we had developed course material to illustrate the principle the linux kernel that ran on the student machines was routinely updated. There was no indication that address space randomisation had been included in the intervening kernel versions until our carefully designed exercise stopped working. Blue-team/red-team exercises will no doubt demand considerable upkeep if the systems, vulnerabilities, attacks and exploits are all to be kept realistically up to date.

If we are to keep our experiments relevant to the real world and thereby to our students, considerable resources will have to be spent in updating exercises, not to mention updating our practical knowledge in order to update the exercises. It should perhaps come as no surprise that we are not alone in experiencing such problems. To quote John Biggs:

> In the days when university classes contained highly selected students, enrolled in their faculty of choice, the traditional lecture and tutorial seemed to work well enough. However, the expansion, restructuring and refinancing of the tertiary sector that began in the 1990s has meant that classes are not only larger but quite diversified in terms of student ability, motivation and cultural background. Teachers have difficulty in just coping, let alone in maintaining standards, and are stressed. [1, p.1]

Modern texts on university pedagogics propose nothing short of an revolution in the organisation and financing structures in higher education in order to address such problems [3, 4]. While we are waiting for that revolution, we hope to make some headway by making the best of the situation.

2 What we have going for us

If we must play the hand we are dealt with, perhaps some of the cards are not too bad.

Modern Pedagogics - Researchers in the field of university pedagogics are encouraging us to cope with the diversity and the multitude of students in a number of ways. Diversity in students and in their learning processes can be addressed by flexibility in the learning environment. To quote Bowden and Marton in their book *The University of Learning*:

> ...students are more likely to adopt a deep approach to learning and are more likely to seek meaning and understanding if they are able to exercise a degree of choice about what and how they learn. It is important that, as much as possible, the learning environment is flexible so that students can exercise such choice and develop as the kind o independent learners we seek. [3, p.267]

In the same book the authors encourage us to shift from a view of our role as teachers to instead be facilitators of learning. This includes not only helping students to learn, but also always being open to learning from our students. This perhaps implies that the greater the number of students, the better are our opportunities to learn from them. To quote John Biggs again:

> Effective teaching means setting up the teaching/learning context so that students are encouraged to react with the level of cognitive engagement that our objectives require. [1, p.56]

IT Security's Fascination - The field of IT Security is perhaps exceptional in that when it comes to its practical elements, most students are strongly self motivated. Having the knowledge of how to beat or break a system seems to have a particular allure. This might not be exactly the *level of cognitive engagement* that we require, but it is at least something to build upon.

Online Collaboration - The latest years have seen the rise of a new kind of social phenomenon on the Internet: Online collaboration and sharing. This phenomenon is so prevalent as to have been dubbed by some Web 2.0 [10]. The trend is evident in many forms, perhaps most notably in the emergence of *wikis* such as Wikipedia [9]. These are sites that contain an abundance of clearly structured information that has come about through the cooperation and general benevolence of Internet users. It is perhaps surprising that even though such a material could be easily vandalised or otherwise spoiled by just about anyone, it has been said to rival the accuracy of the Encyclopaedia Britannica [5]. This phenomenon could work for us in an academic context.

3 The Background

A dedicated IT Security laboratory has been in operation at the department since 2002. It embodies a switched sub-net which is separated from the rest of the department's computer networks by a highly restrictive firewall which in principle only allows http traffic to pass in and out of the subnet. This allows us to relax the security policy that the students are otherwise subject to when using the department network.

The main practical difference between exercises in a security laboratory and those conducted for other computer science courses is that for the most interesting experiments the students must be given, or allowed to acquire, system administrator privileges. This creates problems when the laboratories are shared among students, where each workstation is booked by a single student group for a number of hours at a time. If students are given administrator rights to a workstation they will have limitless possibilities to make that workstation unusable for the next group to use it, intentionally or not. What is more, it should be possible for a project to conducted and saved over several sessions without the risk of other students spoiling the environment for each other. This was solved by giving each workstation a removable hard-drive cassette as its bootable medium. Students are allowed to check out their own hard-drive cassette for the duration of their exercises. The workstations can net-boot under the control of the firewall server, giving the students the possibility to format their disk and to ghost clean images of operating systems to their disks. This environment has subsequently been christened The Sandbox. A number of exercises where devised for use in this environment. These exercises include experiments in Windows with system hardening, keylogging, spoofing, and using Trojan horses. In a linux system, they include experiments with system administration tools that allow for network sniffing, spoofing, vulnerability analysis, and intrusion detection. [2]

Due to the diversity of the students' technical experience there are no specific goals for each of these experiments. Instead, they are formulated as a number of steps that the student should follow and an exhortation that the student should observe and reflect upon what the see. Different students will have differing observational powers, but the idea is that no matter what the background of the student they should be able to make some sense out of what is happening and learn important lessons according to their own individual possibilities.

This was the basis for a one week's practical assignment for most of our introductory courses. Students worked in pairs and were required to hand in a written report where they not only should show that they had completed the assignment themselves, but they should also show evidence of how they had reflected upon their observations. They were exhorted to explicitly tie in their reflections to the theoretical framework that they studied in the course literature and at lectures. Students were not graded according to their results from this exercise (since they were working in pairs, and it is typical in Sweden not to give differential grades for cooperative projects) but given a pass or fail.

The results from these exercises was very varied. Many students reported that the Windows exercises were the most interesting and educational, although as these exercises were designed they are considerably more trivial than the linux based exercises. The linux based exercises were described in simple terms, but no doubt those students who had never before used linux were nevertheless intimidated by an interface so strange to them. Very few of the hand-ins were ever of a calibre that the exercise instructions called for. Reflections were seldom included, and only very vague links to the course material were made if at all. It was clear that students were interpreting the lack of clear goals as a licence to sub-optimize their efforts, and they were putting in far less than the expected 40hrs of laboratory time.

4 The Experiment

Goaded by the problems and encouraged by the possibilities as discussed above, we began a pedagogical experiment in 2005 where the students themselves suggest IT Security projects that they would like to work with. They are now first required to complete the exercise described above, but without the requirement of reflection; they simply have to document their experiments to show that they had completed it on their own. The first assignment now serves primarily as encouragement to get to know the laboratory and the basic tools. This assignment is expected to take no more than 16 hours of laboratory time.

The second part of the assignment, to design and execute an own experiment, should be designed to take 24 hours for each student. To this end they are allowed to form arbitrarily large groups as the experiment may require. No two separate groups are allowed to attempt the same project at the same time. Each project must be proposed in writing and approved by the course staff before work may progress. This is only a cursory process to make sure that the students themselves understand what they are embarking upon and that it is of suitable dimensions for the course. Since the projects are not overly vetted at the start, we spend greater effort on monitoring students' progress in the laboratory in order to catch any unforeseen problems at an early stage, and get the students quickly onto a more productive track.

The first time round, we gave a short list of suitable projects, just to get the ball rolling for those students who could not immediately design an own project:

- Install and test the usability of FreeBSD configured with Mandatory Access Control.
- Suggest a network usage profile (such as family network, or small company) and configure a firewall to suit that profile.
- We have a number of Security products made available to us from Computer Associates, including firewalls and intrusion detection software, together with tutorials. Install and evaluate one of these systems and their supplied course material.
- On the basis of advice on how to secure a system (as found at suitable sites on the Internet), secure the Windows 2000 and Linux systems on your hard drive, and compare the process for the two systems.
- Design exercises for Windows XP equivalent to (or preferably improved from) those you have done for Windows 2000.
- Install and configure a Honey Pot
- Install and test a Tar Pit
- Design a series of attacks for a chosen system configuration, e.g., for a certain version of Windows without all the latest safety packs.
- Install and test 2 different password checking programs.

This basic format was tested on our International Master's Programme in Computer Security two years in a row, 2004-2005. The results were encouraging in that many students could immediately find projects of their own liking. They generally spent more than the stipulated 24 hours per student on their projects, which although it might be considered detrimental to their other studies told us that this was promising from the motivational point of view.

5 Stage Two

Our ultimate ambition with these exercises has been to create a database of reusable security related experiments that students could pick and choose amongst as well as add to. This puts a greater demand on the final hand-in, i.e. the documentation, than would normally be required on such a course. We are more intent on the students learning a useful lesson from their experiment than worrying about finer details of its documentation. However, when students are running an experiment based on the documentation written by fellow students, whether finer details are included or not can mean the difference between effective learning and total frustration. If the course staff were responsible for checking the documentation in detail and requiring multiple complementary hand-ins until it reached the required level the whole process would become prohibitively costly.

As an alternative strategy we turned to the idea of wikis [8]. If the communal spirit can result in such a full and well structured material as the Wikipedia, we hope that the same could contribute to the gradual expansion, update and refinement of a student driven database of security exercises. In the Autumn term of 2006 we transferred some of the more interesting and better documented projects to a locally run wiki site. Students could now suggest their own projects by creating their own wiki page and submitting it for approval, or else they can choose an existing project from the wiki, mark it as booked, and then repeat the project with an aim to testing and improving its documentation.

We believe that some standardised formatting for these wiki pages must necessary be imposed to help the students to quickly decide on whether a project is of a suitable type and size to suit their level of abilities and their group size. As yet, it seems sufficient to impose a few standard header fields, such as project prerequisites and expected person hours to complete.

6 Preliminary Indications

Thus far we can say that our pedagogical experiment shows great promise. The database currently comprises of some 46 experiments, even if some are very close to another in their and one or two have been repeated in both Swedish and English with separate documentation. Initial worries that this strategy would result in chaos have proved to be unfounded. It seems that the students themselves are good judges of their own abilities and do not in general pick projects that get them into trouble. There is a proportion or projects that do fail to reach their original goal, but with expedient management from the course staff most of these worst effects have been avoided without it costing the students a great deal of frustration or the course staff a great deal of time.

At the time of writing, the second group of students to use wiki based documentation are deep into experimentation. The difference for this group compared to the previous ones is that they are heavily relying on texts written by previous students (sometimes foreign students with poor language skills in English) and it is already clear that they are developing an understanding of what is required of the documentation to make a project easily repeatable. A number of groups have expressed their prime goal to be to improve the documentation. As one of the explicit goal of the introductory course is that students should be able to communicate about IT Security after completing the course, this is a very positive result. It is, however, only relatively positive; we have found the level of ambition in the clarity and purpose of the documentation is mostly disappointing. Students are still largely motivated to show that they have done enough work for a pass, rather than to write the kind of documentation that they themselves would like to read.

In terms of learning from the students, this has been a very valuable project. From the course staff point of view many projects have turned out to be surprisingly easy. Some have been innovative. Some are surprisingly difficult, such as experiments in propagating worms and viruses, which have seldom worked. Without this eager student workforce the course staff would never have had the time to make the same discoveries.

We have for the most part been liberal and lenient in what projects we have allowed and what documentation we have passed. It has however been necessary to employ strict requirements in one particular aspect, i.e., ethics. Some projects have been suggested without due regard for the safety and security of the environment in which the experiment is to be conducted. This is despite considerable emphasis being put on issues of policy and ethics during the lecture series. Students who wish to conduct potentially dangerous experiments, such as building a simple virus or sniffing a wireless network, are required to give detailed specifications of the measures they will take to ensure that they minimise the risk to their environment. This at least is evidence that practical experiments are useful for driving home important lessons from the lectures, such as those on ethics, that are otherwise too easily missed.

7 The Future

A fuller evaluation of this experiment will be conducted after completion of the 2007 spring term course, where the normal student course evaluation will be augmented with more detailed questions on their experiences from the assignment. A thorough review of the current state of the experiment database will also be conducted. The preliminary results are however promising enough to suggest several ways to develop the ideas presented here.

The size and complexity of the current experiments has been limited by the 24hour per student limit. We would like to encourage more ambitious projects that can span a longer time period. This could be done once the exercises in the database have a stable enough core by the creation of a special course in practical experimentation. Students would be required to amass a number of hours and educational credits as related to the values in the experiment headers, to gain study credits.

We are enthusiastic to the idea of using virtual machines in order to prepare environments where the experiments are ready to run. It would also be very practical if one could run virtual machines as virtual security laboratories, thus liberating the student from the requirement to stay within the confines of our laboratory. We doubt however that these environments can be made secure enough to be confident that they will protect the systems they run within. Students should not have to become experts in the configuration and running of virtual machines in order to run security experiments.

A database of security experiments will invariably contain a certain amount of information that would be of interest to persons with ill intent. Though we hold the view that the experiments themselves are not malicious, and while we make every effort to ensure that our students use any knowledge gained for altruistic purposes only, it is a sensitive issue whether the department can or should make the database publicly available. For the sake of the material itself it would be most advantageous to allow the Internet in general to have access. Until that time, a happy compromise would be if we could establish cooperation between other schools of higher education so that such experiments could be shared. A network of secure VPN connections might also allow for the interconnection of similar security laboratories, which could open up exciting possibilities for not just shared experiments, but cooperative experiments.

References

1. Biggs, J.: Teaching for Quality Learning at University Open University Press, Slough, UK, 2003.
2. Blomquist, N., Nilsson, P., Peris, A.: Praktisk IT-säkerhetsutbildning - specifikation av utrustning, moment och administration, Master's Thesis No. 02-74. DSV, Stockholm University, 2002. In Swedish.
3. Bowden, J., Marton, F.: The University of Learning. RoutledgeFarmer, London, 1998.
4. Laurillard, D.: Rethinking University Teaching. Routledge, NY, 1993.
5. Nature: Internet encyclopaedias go head to head, March 2006, `http://www.nature.com/news/2005/051212/full/438900a.html` [visited 20070211]
6. Viega, J., McGraw, G.: Building Secure Software. Addison-Wesley 2002.
7. Vigna, G.: Teaching Network Security Through Live Exercises, in Proceedings of IFIP TC11/WG11.8 Third Annual World Conference on Information Security Education (WISE3), California, 2003.
8. wiki.org: What is Wiki?, `http://www.wiki.org/wiki.cgi?WhatIsWiki`, June 2002, [visited 20070211]
9. Wikipedia: `http://www.wikipedia.org/` [visited 20070211]
10. Wikipedia: Web 2.0. `http://en.wikipedia.org/wiki/Web\_2` [visited 20070211]