# Teaching Data Security at University Degree Level

Maximillian Dornseif, Felix Gaertner, Martin Mink, Lexi Pimenidis

December 20, 2004

There is a general consensus that courses on data security at university degree level should be research-oriented and teach fundamentals of the field, i.e., items of long-term knowledge in contrast to technology-oriented system knowledge. Unfortunately, this consensus often results in courses that are either too theoretical or are outdated with respect to current developments in security technology. We argue that combining research-oriented approaches to data security education at universities with practical and system-oriented elements is no contradiction. We present the outline of a university degree curriculum that combines theoretical and practical topics in a novel way. Our approach differs from previous curricula by a unique combination of (1) extensive practical excercises and (2) research-oriented treatment of offensive techniques in data security.

**Keywords:** university degree curriculum, practical elements, computer forensics, summer school, offensive techniques

## 1 Introduction

At university degree level, it is a rule of good academic practise to teach long-term methodological knowledge instead of short-term system knowledge. In the area of data security, this has resulted in university curricula which either tend towards theoretical topics (like cryptographic protocols or formal modeling of security) or towards practical topics highlighting defensive strategies (e.g., access control techniques, firewalls and VPNs). Data security, however, is a field which is rapidly changing. The new developments like the security threats in Web-based systems (e.g., cross-site scripting) or the dangers of so-called botnets are often neglected. This leaves university graduates with only faint ideas of the security threats they will face in their professional career. Moreover, a typical computer science graduate, even if she has specialized in data security, usually has very little practical experience with the way *real* systems react in the presence of malice.

We argue that practical experiences with real security failures should be a central part of university degree level education. Furthermore, our main claim is that the quality of data security professionals with university degree can only be improved if *offensive* aspects like writing exploits or network sniffing are integrated into the curriculum. From our experience, this combination of practical experiences and offensive techniques yields graduates that can both (1) improve the level of security in non-academic contexts and (2) perform high-quality academic research in the advancement of security engineering principles. We believe that offensive techniques are central to better understand the ways in which security systems fail. And there is an increasing awareness, that understanding offensive techniques raises the overall level of security instead of lowering it [6, 3].

In this paper we present the outline of a two-semester university degree curriculum that to our mind improves the state of the art in security education. It consists of two semesters:

- The first semester has three elements: (1) a (traditional) lecture on *data security techniques*, (2) a lecture on *computer forensics*, and (3) a *research seminar* on current trends in computer security where students give a presentation.
- The second semester consists of an extensive *practical lab session* in which students apply offensive and defensive techniques within an isolated test network. The final part of the semester is a two or three week *Summerschool* in which advanced attacking techniques are trained and analyzed.

We report on our experiences with this curriculum which is being developed at a major German engineering university.

The guidelines which lead us to develop our curriculum in the described way can be formulated as four *principles of security education* at universities:

1. Students should collect extensive practical experiences with current security technologies.
2. Students should learn from the failures of defensive security technologies.
3. Students should learn how to apply offensive security techniques.
4. Teachers should trust their students.

Principle 1 addresses the lack of practical experience which graduates often have today. Principle 2 takes a new look at those techniques that are part of standard security courses or textbooks today. Principle 3 complements this by explicitly teaching ways to search for and exploit security vulnerabilities in current systems. Principle 4 is a basis for educating students in ethics of the security profession, a point made necessary by principle 3.

**Related approaches.** The most traditional part of the proposed curriculum is the first semester basic lecture on data security. Similar lectures exist in almost all prestigious universities around the globe whereas lectures on computer forensics are not as widespread,

at least not in Europe. Also, the combination of lectures, seminars and extensive practical exercises focussing on offensive techniques which we propose in this paper has not been implemented at any university we are aware of.

There have been some good attempts to incorporate practical elements into university degree courses on data security. The computer science department of Darmstadt University of Technology, Germany, regularly runs a so-called *Hacker Contest* for several years [12]. The Hacker Contest is a lab course in which students form teams that have to set up systems and then use common exploitation techniques to attack the systems of the other teams, analyze attacks to their own systems and increasingly deploy stronger defense measures. The University of Magdeburg, Germany, also offers a similar lab course and has joined efforts with Darmstadt in that students from Darmstadt perform penetration tests of systems deployed in Magdeburg and vice versa. In the military education, one can find similar examples of offensive lectures, for example [15].

Several projects have pioneered the use of offensive techniques as teaching concepts but none of them has treated offensive techniques in a research-oriented way, as it is done in the Summerschool of our curriculum. So called *Wargames* and *Capture-the-Flag* contests have a long tradition among security enthusiasts. In Wargames the organizer creates a set of challenging scenarios of increasing difficulty which have to be solved by the participants. Challenges usually are modeled somewhat after the problems an attacker faces when attempting a system penetration. Typical Wargames can be found at [4, 10]. Slightly more competitive than Wargames are so called *Capture-the-Flag* or *Deathmatch* contests where teams battle against each other over the control of a network. Most famous is probably the Root-Fu contest in the United States [8] and the CTF contest of the UCSB, in which several educational institutions spread across the United States battle against each other [14, 13].

The Information Technology and Operations Center at the U.S. Military Academy West Point has a curriculum which also teaches offensive information security techniques. The center also organizes a yearly *Cyber Defense Exercise* which has similarities to the Capture-the-flag contests. U.S. authorities with an information security education branch like the United States Military Academy, the United States Air Force Academy and the Naval Postgraduate School participate in these exercises. Machines maintained by the participants are attacked by the NSA 92nd Aggressor Squadron – Land Information Warfare Activity over the course of several days and participants have to counter these attacks [5, 11].

**Outline.** In the remainder of this paper we first give details on the basic lectures taught in the first semester of our curriculum (Section 2). We then give details about the agendas of the practical lab session (Section 3) and the Summerschool (Section 4). We conclude in Section 5.

## 2 Lectures

In this section we briefly outline the two main lectures taught in the first semester of our proposed curriculum. We first present the basic lecture on Applied Computer Security and then the lecture on Computer Forensics.

### 2.1 Applied Computer Security

The lecture "Applied Computer Security" is maybe the most classical part of the curriculum presented in this paper: It is a standard lecture (4 lecture hours per week) in which basic concepts of computer security are explained to the audience. The lecture can be characterized as being a merger of four very different textbooks on the area of computer security: the books by Gollmann [9], Amoroso [1], Garfinkel, Spafford and Schwartz [7], and Anderson [2].

The textbook by Gollmann [9] is often considered to be the book with the broadest range of topics, reaching from authentication in Windows NT to security certification, on the least number of pages. Its conciseness is, however, also a downside since specialized aspects such as theoretical models (treated by Amoroso [1] in more detail) or in-depth case studies (treated by Anderson [2]) need space. All three books are stated as recommended reading, and for individual topics which are treated in the lecture pointers to the relevant chapters in the books are given.

After discussing basic terminology issues, we chose to follow the path laid out in the book by Garfinkel, Spafford and Schwartz [7], i.e., the course first looks at the security concepts which are present in a standalone UNIX or Linux PC. The list of topics covers among others passwords, authentication techniques, access control, root priviliges, file permissions, cryptography basics, backups, software patches and physical security. The second part of the course covers network and Internet security issues, e.g., securing TCP and UDP services, network based authentication systems, network filesystems and secure programming techniques.

Whenever suitable, common vulnerabilities and attack techniques are discussed to focus on weaknesses of current technologies. For example, dictionary attacks on passwords are discussed in the lecture on password-based authentication, or buffer-overflow attacks are discussed in the context of secure programming techniques. From our experience it is better not to separate attack techniques from defensive techniques because it gives students a better understanding of the tradeoffs in security technology and makes the lecture less boring. Apart from the required reading and some discussions during the lecture there are no lab sessions and no homework assignments. The lecture is meant as a "mind opener" to a broad range of aspects of computer security.

### 2.2 Computer Forensics

The course on computer forensics is taught with two lecture hours per week and thus has half the size of the previous course. Forensics or forensic science is the application of science to questions which are of interest to the legal system. Classic computer forensics

is the gathering, interpretation and presentation of evidence found on computers. Since the aim of forensic science is to supply services to the legal system, forensics are very dependent on the kind of legal system they work with. Unfortunately nearly all literature on computer forensics originate from places where the criminal justice system stems from the common law which is very different from the German legal system. This and the fact that in the German legal community there is up to now very little experience in using computer forensic evidence in court cases makes it difficult to give students firm guidance on how to interact with the legal system and how to testify as an expert witness in front of a court.

In our teaching we broaden the definition of computer forensics: We understand computer forensics not only as a tool for the legal system, but also as a tool for understanding security. Sound engineering principles dictate a tough analysis of failures to learn the workings of a system and avoid subsequent failures of the same kind in the future. We define computer forensics as the attempt to reconstruct the events which lead to a security policy violation in an information security system. Thus computer forensics also includes the analysis of security incidents to learn the tools, tactics and techniques of the attackers and to gather facts needed to improve security in the future.

In the light of the lacking experience of interaction between the German legal system and computer forensics and our determination to base further research on information gained via forensic analysis we put much effort in teaching students to adhere to sound scientific principles when conducting forensic examinations. Exact documentation of all steps taken and well proofed strategies to minimize the likelihood of alteration of the data gathered or the system examined lead to reproducible results meeting rigid scientific standards and thus having a good verisimilitude of being accepted by courts and other parts of the legal system.

The computer forensics lecture and accompanying exercises aim at providing students with the necessary knowledge to understand evidence on computers at a very deep level. A big part of the lecture consists of deepening the topics from operating system and systems programming courses in areas of specific interest to forensics. These are namely networking, process management and filesystems with a strong bias to the latter. Based on a deep understanding of how relevant parts of information systems work, students learn how to extract and interpret evidence from such systems and to evaluate the validity of that information gathered. We aim at giving the students a fundamental view on how the extraction of evidence from IT systems works, enabling them to conduct forensic analysis without anything but the most basic tools. Drawing from the strong engineering skills put forward by our university our students should have the ability to develop tools they need to make their forensic analysis more swift whenever they are in need of such tools. Ready made software tools are only covered briefly in the lectures since we assume that the fundamental knowledge acquired during the class should enable students to quickly understand the forensic tools available on the market.

The exercises accompanying the lecture aim at giving the students opportunity to experience different forensic techniques themselves. They cover a wide spectrum of tasks ranging from IP-backtracking to reverse engineering of unknown binaries. To allow students to gather first hand experience with data capture and analysis we acquired a

large amount of pre-used hard disks and ask the students to image and analyze them. We also try to connect lectures and exercises to recent events, e.g. the trustworthiness of voting machines and ex-post analysis of the equipment used in an election.

The course on computer forensics seems to be popular among students. Even while the course is not mandatory, the exercises are neither graded nor a requirement and the course usually will not finish with an exam we see constant attendance of about 120 students and a lot of dedication in the exercises.

# 3 Practical course and seminar

First, we present the seminar *Hacker Seminar*, a theoretical course, but with practical elements: students prepare a presentation on an aspect of IT security with a practical demonstration. Then we give details about the practical lab session *Hacking Lab*.

## 3.1 Seminar Hacker Seminar

Somewhat more theoretical than the practical course Hacking Lab, we offered a seminar on typical hacking techniques and security measures. In the course of the seminar, teams of two students had to research up-to-date information about a security related topic and prepare a talk on their results. The talk had to present the theoretical background and the practical relevance of a given topic, combined with a short demonstration and a discussion at the end. The target of this lecture is to show typical weaknesses in IT security, how they can be avoided, and raise awareness on the topic. Some examples for the topics covered are: Spam, Buffer Overflows, Root Kits, Social Engineering or Attacks on authentication systems.

The talks were given to a public audience to allow students from all grades to participate and gather new knowledge or update their knowledge on IT security. Due to the big general student interest the talks were indeed highly frequented.

## 3.2 Practical course Hacking Lab

The Hacking Lab is a practical introduction to defensive as well as offensive computer security measures for a small number of studente. Thus we extend the students' existing theoretical knowledge base by letting them experience failures in real systems under their control in the presence of malice.

Goal of this course is to give students the opportunity to learn real life security in a controlled environment. We think the best way to do this is by getting to know both sides: as an administrator of a computer learn the defensive measures, as a hacker learn the offensive measures. The computers used are inside an isolated test network, giving the students the possibility to try attacks and getting attacked without implications to the outside (Internet, Intranet).

Goal of this course is to give students the opportunity to learn real life security in a controlled environment. We think the best way to do this is by getting to know both sides: as an administrator of a computer participants learn the defensive measures, as a

| Hardware | Operating System | Services |
|---|---|---|
| PC | Debian Linux | DHCP-Server |
| | Redhat Linux | NIS- and NFS-Server |
| | SuSE Linux | Apache webserver with PHP and DB |
| | Windows ME | |
| Laptop | Windows 2000 Server | MS IIS with DB |
| | Windows XP | IRC-Server |
| SUN Sparc | Debian Linux for Sparc | FTP- and Samba-Server |
| | Solaris | sendmail |

Figure 1: HW and SW used in Hacking Lab 2004

hacker they learn the offensive measures. The computers used are inside an isolated test network, giving the students the possibility to try attacks and countermeasures without implications to the outside (Internet, Intranet).

To represent a realistic scenario we chose a business setting. Therefore we divided the students into four teams of two or three, each team representing a department of a fictitious company (e.g. accounting, controlling, IT, marketing, personell). The departments have to offer each other services (e.g. file server, databases, webservices). Each team administrates several computers, that are connected via a network. The students install the systems (operating systems and services) and then on one hand try to secure their own systems, on the other hand try to find vulnerabilities in the other departments' systems. To complete the realistic scenario we used different hardware and different operating systems. For a list of hardware, operating systems and services see figure 1.

To ease their offensive task, the initally chosen software is at least one or two years old and has several well known and exploitable vulnerabilities. We have chosen to start with older software because the topic of this course is not to find new security bugs, but rather to learn how to react to and live with software in the presence of vulnerabilities. Remember that the software was originally not known to have these problems, so it was deployed in productive systems the same way as today's software is. Although there are less known exploits for today's software, it is common conception that every larger piece of software is vulnerable to some extent. Thus we don't consider the use of older software a restriction but rather a help. It additionaly demonstrates how easy it is to break into systems that are not closely watched and carefully patched.

Usually it takes only a short time from the beginning of the course to the first results. As we mentioned before, it is not the main purpose of this course to attack vulnerable hosts, so the actual work is to be done after a successful attack: both the attacker, as well as the defender, have to analyze and document each incident. We demand to list and explain the cause of the incident, the analysis that has taken place before the exploit, the actual attack and the clean-up afterwards. At this point, the students have to show their understanding of what happened.

Time is divided into phases with a certain task. During the phases students work freely

| Phase | Duration | Description | Goals |
|---|---|---|---|
| Installation | 2 weeks | Install OSs and services | Installation and administration |
| Attacks I | 3 weeks | Attacks, sniffer, fingerprinting, ... | Get to know the tools, the network and computers, and vulnerable software |
| Secure systems | 1 week | Secure systems against attacks | What measures have to be taken to secure a system? |
| Attacks II | 2 weeks | Continue with attacks | What has changed by securing the systems? |
| Configure Firewall | 1 week | Create a ruleset for the firewall | Which packets, ports and protocols should be blocked? |
| Attacks III | 2 weeks | Attacks against hosts behind firewall | Is it still possible to attack other hosts? Attack the firewall itself? |

Figure 2: Schedule of the Hacking Lab 2004

on their task. To let them experience the difference between an insecure and a secure environment, the general level of security is raised over time. At the beginning, only unmodified and unpatched operation systems and tools at least one year old are allowed. Also, every group was required to offer a login for local shell access on their machines. In a second step, the students may apply patches and general security measures to their systems as defensive measures. Finally, the training network of the deployed computers is split and a firewall is planted between the hosts. The configuration of the firewall is done according to the feedback from the trainees. Each phase concludes with a meeting where the teams present their experiences and discuss the incidents. Figure 2 depicts the schedule of the Hacking Lab in 2004.

We expect the students to have gained knowledge at the end of the course of how to handle a server that has to provide services in an environment where neither the software nor the users can be trusted. Additionally we force the students to analyze what actually happened during the execution of an attack and what measures could have been taken to avoid it.

### 3.3 Experiences and outlook

Both the seminar and the practical were very popular among the students. For the seminar around 100 students applied for only 16 positions, for the practical course somewhat less (which is due to the fact, that not all students need a practical course).

The talks of the seminar attended up to twice as much guests than participants – other students interested to learn more about security themselves. Asked why participants and guests attended the talks, some of the answers we got were “gain an overview”, “make own computer secure”, “the demonstrations”or “learn about secure programming”. The seminar was evaluated by the participants at the end: 80 % said they had learned a lot

in the course, and all participants gave the course a good rating.

With the practical we had some problems to cope with. For example installing old software (Windows ME) on new hardware (laptop) was a problem. Also, installation of software on "exotic" hardware (SUN) was tricky and finding out that some computers have a 10-Mb-only network card (which doesn't work with a 100-Mb-only hub/switch) was difficult. Another problem arose from the fact that we had giving groups shell access on unpatched remote machines which are unpatched. The next time we will refrain from doing so — at least not from the very beginning — because it was too easy to gain root priviliges.

For future lab sessions we will increase the number of participants (probably 16 in four teams of four). The schedule will vary to incorporate other topics relevant to security like mobile security (WLAN, Bluetooth), forensics, IDS, routers, hardware hacking (e.g., X-Box), biometric devices or privacy techniques. Additionally we plan to deploy a so called *Game Server*, a dedicated computer in the lab network, which regularly probes the adherence to tasks given to the teams (e.g. making a file available on the webserver, that is just downloadable from a certain computer). This will require them to configure some holes in their systems in a secure way (a quite realistic scenario).

# 4 Summerschool Applied IT Security

Our most condensed effort so far took place in the Summerschool *Applied IT Security*. This three week effort in September and October 2004 was meant to be an intensive course based on the principles outlined above. Students were given the opportunity to extensively induce and study failures in security systems.

A secondary goal was to get participants of different skill levels to cooperate and thus foster knowledge transfer. We especially wanted to have more experienced researchers like PhD students to team up with less experienced students. We hoped that this would help to introduce students to the scientific approach and possibly interest them in information security as a science.

## 4.1 Organization of the Summerschool

The intended audience were students and young scientists from Europe with a profound interest not only in IT security but also in offensive techniques in relation to IT security. We also expected a high level dedication of technical skills: participants were expected to work hard during the Summerschool and to use an extreme wide array of tools and techniques in the lab sessions.

Since these traits could not be checked like simple requirements in an application form we decided to use some kind of scare tactics to deter less determined students and give others the opportunity to show creativity. For example we requested a list of publication in the application form. While some participants could offer a page long list of scientific publications we were more interested in the tactics used by others. So some applicants put advisories or even newsgroup postings in their applications – we generally preferred this over no entry at all.

| Day | Lecture 1 | Lecture 2 | Lab |
|---|---|---|---|
| | 8:45-10:15 | 10:45 - 12:15 | at least until 18:30 |
| Week 1 | | | |
| 1 | Introduction | Hardware Security | Hardware / Wargames |
| 2 | Web Applications | Web Applications | Web Applications |
| 3 | Buffer Overflows | Other Programming Errors | Exploiting Overflows |
| 4 | Advanced Exploitation | Networking | Network Programming |
| 5 | Sniffing: Layer 1 & 2 | Spoofing, DoS & DDoS | Spoofing |
| Week 2 | | | |
| 1 | Network Topology | Applications Fingerprinting | Network mapping |
| 2 | Bluetooth | Wireless Attacks | Wardriving |
| 3 | Hidden Data | Honeynets | Wardriving |
| 4 | Introspection | Projects | Projects |
| 5 | Projects | Projects | Projects |
| Week 3 | | | |
| 1 | Misc. forensics | Disk Forensics | Forensics |
| 2 | Disk Forensics | Disk Forensics | Forensics |
| 3 | Malware | Unix infection | Honeynets |
| 4 | Excursion | Excursion | Excursion |
| 5 | Wargame | Wargame | Wargame |

Figure 3: Schedule of the Summerschool 2004

We did only very low profile advertisement for the Summerschool. Besides announcing it at the laboratory's Web page and mentioning it to friends and collegues we made an announcement on a mailing-list dedicated to penetration testing. Since this already resulted in more than 50 applications we refrained from doing further advertisement.

Then we selected 16 students on the base of prior experience, expected willingness to work hard and interest in scientific work. One of those students was from Turkey, four from Great Britain (three of them members of Professor Ross Anderson's Security group), one from The Netherlands, and 10 from Germany (Berlin, Cologne, Aachen and Gelsenkirchen). Two of the selected students informed us only a few weeks before the start of the Summerschool that they would be unable to attend so the final number of participants was fourteen.

For the lab sessions we provided a room where the students could use their own laptop computers and which held a lot of obscure hardware and complex network infrastructure to experiment with. To foster informal communication between students, we converted an office to a coffee hall with sofas, an overall relaxed ambience, and a choice of snacks and refreshments. Finally, there was a plain room for temporary use by students which had the feeling they need concentration for specific tasks. The housing was left to the students themselves and their choices showed a very wide variety of solutions ranging from the universities' guest-house to the local camping ground.

The schedule during the Summerschool was demanding: Lectures started at a quarter

to nine in the morning and covered two topics until noon. After lunch, the lab session started, during which students applied the techniques learned in the lectures and developed them further. While some of the lab sessions took a very guided approach asking students to solve specific tasks, others were more general up to the point where students were just asked to apply the knowledge gathered in the morning. For an overview of the schedule see figure 3.

Each lab session was interrupted by a so called *coffee table talk* where external speakers gave a short statement related to a topic of their interest or a short introduction to some project they work on. The coffee table talks were intended to broaden the view of the participants and to see problems not only from the security perspective and to get a focus on problems being faced in the real world. Speakers included people from industry, e.g., from Microsoft and Pixelpark up to German Postbank and the TÜV Rheinland. But there also was participation by other groups like the Chaos Computer Club (CCC) or the "Verein digitale Kultur" which covers artistic expression via digital means and various academic research groups. Topics covered in the coffe table talks ranged from computer-ethics, civil liberties in relation to the Internet, phishing attacks from a banks perspective up to highly technical subjects like attacks on memory management and XML security.

A typical day ended with a meeting: usually around six in the evening everybody presented his work of the day. But students often stayed through large parts of the night to develop their projects further.

### 4.2 Experiences and outlook

While the lab sessions with less guidance lead to some very impressive results, especially the less experienced students felt that they had a better learning experience in the lab sessions with more instructions. Nevertheless we found several vulnerabilities and flaws in software and systems. All of these weak points were reported to the authors respectively the operators. In cooperation with these persons, the holes were fixed.

As a result of the Summerschool participants made six submissions to the Chaos Communication Congress, an annual meeting of hackers in Germany, organized by the Chaos Computer Club. All six were accepted for presentation.

The biggest lesson we learned is: Practical aspects of security and exploitation of real systems are fun for students. After the students had exploited their first buffer overflow, they were very enthusiastic and worked hard to learn more.

All in all the event was an immense drainage on everybodys power but also a huge success. It is already planned to be repeated it in 2005.

## 5 Conclusions

The main statement of this paper is that combining research-oriented approaches to data security education at universities with practical and system-oriented elements is no contradiction. In this paper we sketched a two-semester university degree curriculum that to our mind improves the state of the art in security education: A mixture of lectures

and a seminar in the first semester provides students with a basic background in computer security terminology and techniques. In the second semester students extensively gain hands-on experience in state-of-the art offensive and defensive techniques. Existing approaches either lack hands-on experience or research-oriented treatment of offensive techniques.

We believe that offensive techniques are central to any form of professional security education. Taught in the right context, only offensive techniques offer the potential to gain an advantage in the race between attackers and defenders in secure computing. They also have a motivating paedagogical impact, since offensive techniques can be used in game-like situations. As can be expected, it is rather difficult to keep the course material up-to-date since security technologies change too rapidly. However, we feel that this task is a challenge we need to meet to improve the general understanding and master the difficulties in today's security domains.

## References


[1] E. Amoroso. *Fundamentals of Computer Security Technology.* Prentice Hall, Englewood Cliffs, NJ., 1994.

[2] R. J. Anderson. *Security Engineering — A Guide to Building Dependable Distributed Systems.* John Wiley & Sons, 2001.

[3] I. Arce and G. McGraw. Guest Editors' introduction: Why attacking systems is a good idea. *IEEE Security & Privacy*, 2(4):17–19, July/Aug. 2004.

[4] Digital Evolution. Homepage "digital evolution". `http://www.dievo.org/`. Accessed 2004-10-15.

[5] R. Dodge, D. J. Ragsdale, and C. Reynolds. Organization and training of a cyber security team. In *Proceedings of the 2003 IEEE International Conference on Systems, Man & Cybernetics*, 2003.

[6] D. Farmer and W. Venema. Improving the security of your site by breaking into it. Usenet Posting to comp.security.unix, 3. Dec. 1993.

[7] S. Garfinkel, G. Spafford, and A. Schwartz. *Practical Unix & Internet Security.* O'Reilly & Associates, Inc., third edition, 2003.

[8] Ghetto Hackers. Homepage "root-fu". `http://www.ghettohackers.net/rootfu/`. Accessed 2004-10-15.

[9] D. Gollmann. *Computer Security.* John Wiley & Sons, 1999.

[10] Hack this page. Homepage "hack this page". `http://www.hackthispage.tk/`. Accessed 2004-10-15.

[11] W. Schepens and J. James. Architecture of a cyber defense competition. In *Proceedings of the 2003 IEEE International Conference on Systems, Man & Cybernetics*, 1998.

[12] M. Schumacher, M.-L. Moschgath, and U. Roedig. Angewandte Informationssicherheit: Ein Hacker-Praktikum an Universitäten. *Informatik Spektrum*, 6(23), June 2000.

[13] UCSB. Homepage "UCSB Capture The Flag". `http://www.cs.ucsb.edu/~vigna/CTF/`. Accessed 2004-11-05.

[14] G. Vigna. Teaching network security through live exercises. In C. E. Irvine and H. L. Armstrong, editors, *World Conference on Information Security Education*, volume 253 of *IFIP Conference Proceedings*, pages 3–18. Kluwer, 2003.

[15] G. White and G. Nordstrom. Security across the curriculum: using computer security to teach computer science principles. In *Proceedings of the 19th International Information Systems Security Conference*, pages 519–525, 1998.