Is Teaching Hacking in Academia Ethical?

Felix Gröbert, Tim Kornau,

Ruhr University Bochum
Faculty for IT Security
Universitätsstr. 150
44801 Bochum, Germany
felix@groebert.org, opti@openbsd.de

Lexi Pimenidis

University of Siegen
Hölderlinstr. 3
57075 Siegen, Germany
pimenidis@fb5.uni-siegen.de

Abstract. We claim that the method and the content of teaching hacking in academia is well-suited for developing a student's mindset in more regards than solely learning IT security. We describe experiences with teaching offensive IT security topics at university degree level.
We then discuss from several points of view, if teaching offensive methods in academia is ethical and come to the conclusion that the potential risks are well worth the advantages. Finally we also propose a conjecture that offensive methods are likely the best method for working in the area of security for real computer systems.

1 Introduction

Hacking is an activity which is applied to many areas of work: Academics, Free Software Development, Hardware Manipulation and Information Security. One definition is given in the 1981 version of the Jargon File [1]:

> *1. A person who enjoys learning the details of programming systems and how to stretch their capabilities, as opposed to most users who prefer to learn only the minimum necessary.*
> *2. One who programs enthusiastically, or who enjoys programming rather than just theorizing about programming.*
> *4. A person who is good at programming quickly. Not everything a hacker produces is a hack.*
> *5. An expert at a particular program, or one who frequently does work using it or on it; example: A SAIL hacker'. (Definitions 1 to 5 are correlated, and people who fit them congregate.)*
> *6. A malicious or inquisitive meddler who tries to discover information by poking around. Hence 'password hacker', 'network hacker'.*

The Jargon File also reports that the term hacker was used in conjunction with 1950's radio engineers, which endorses the performance of a hacker as a creative, tinkering person.

[1] http://www.catb.org/jargon/oldversions/jarg1-81-MM-DD.txt

In general, and in accordance with the humanist ideal of liberal education from Wilhelm von Humboldt[2], it is the goal of universities to develop individuals, which have an autonomous mind. Students should not be taught knowledge as an end in itself, but rather to develop their personalities for a higher meaning in life.

As we can see, teaching students not the tools and methods of hacking, but rather the mindset of a hacker clearly satisfies the goals of Humboldt. In accordance with this noble goal, this work targets to describe methods on how to teach students to develop a hacker's mindset.

Unfortunately, teaching at university degree did not went along these lines in the area of IT security. The landscape of IT in general, and hence also IT security, changed rapidly within the last few years. For this reason lecturers tried to avoid teaching short-lived information, like e.g. knowledge about current security issues in real systems. The result was lectures filled with so called "eternal knowledge", like cryptographic mechanism or Firewalls. Concepts which were applicable in the 1980s and, admittedly, still have validity.

However, we are not convinced that this challenges students hard enough to actually start thinking. It is also hardly imaginable that this can be reached by conveying theoretical concepts. We therefore examined, to which extend we have to combine the more theoretical approach together with more practical oriented tasks; using each with its advantages to achieve the desired results.

In our approach, both parts play vital roles and stimulate each other: The *theoretic background* is needed to give the student the skills to adapt their knowledge to new problems and question setups quickly. It also makes them independent from tools crafted by others. Other advantages include that students are able to immediately recognize changes in information technology and analyse them according to security deficiencies.

The *practical background* is needed to address the requirement of students to *act quickly* and *efficiently* in any given environment[3]. This includes to be familiar with the usual tool chains on various operating systems and have a firm understanding in a range of programming languages and operating system interfaces. Without this experience students will hardly be able to make correct judgements when under pressure of time and budget, as given in their later lives.

Conventional academic interpretation of information system security comes from a defensive point of view. Certainly, the act of attacking IT systems is only pre-requisite for a very limited amount of jobs. On the other hand, taking the role of an attacker creates a situation which allows for a significant degree of freedom and creativity: while a defender of an IT systems has to regard *all* vulnerabilities and close them one by one, an attacker can *choose* which way to take for compromising systems. The student will be faced with a new degree of

[2] http://de.wikipedia.org/wiki/Humboldtsches_Bildungsideal (German)

[3] Immanuel Kant argues that using reason without applying it to experience will only lead to illusions, while experience will be purely subjective without first being subsumed under pure reason. (Kritik der reinen Vernunft, first published 1781)

freedom and challenges. Having to choose himself which way to take, learning that some ways are more costly than others will result in valuable experiences.

IT security also offers a *unique* experience in the area of computer science: it is a field where two human entities act with opposing interests. The defender of an IT system tries to fend of attackers. Hence, this area is best suited for motivational teaching: to overcome an interactive adversary and beat him with a clever strategy is known to be one of the best drivers for motivation.

In addition, knowledge about adversary patterns and the experience how an adversary thinks and works, will make the student able to detect, prevent and develop countermeasures against today's and upcoming security threads more effectively. This is stimulated and enforced by teaching the circumstances and inner workings of vulnerabilities rather than utilizing out-of-the-box tools. Ultimately, this switch in doctrine brings *liberation* to the students' minds, as envisioned before.

Another advantage of IT security is its persistence: virtually all of today's software technology is defective. Even software that has been thought to be safe might be vulnerable again due to changes in the context[4].

2 Case Studies

In this section we will cover a number of case studies, where universities introduced courses on offensive IT security methods.

2.1 RWTH Aachen University

The RWTH Aachen University has a history of teaching offensive methods in IT security since 2004. The basic principles of the course stayed the same, even though the actual content changed over time (and usually its amount increased). Central ideas included to give the students a broad overview on contemporary IT security issues and point out weaknesses which were not "in the book", or at least not widespread. This illustrates the ubiquity with which security can be compromised. While this also requires more effort in preparing the lectures it makes it impossible for students to simple use Internet searches for retrieving ready tools which do the job.

For the practical part, a virtual subnet comprised out of virtual machines is used[5]. Vulnerabilities discussed in the lectures can then be tried by the students as part of their homework assignments with bonus scores for elegant, short, or otherwise unique solutions.

Networks are covered first, as they themselves have a long history of security problems, some of them being inherent to the design of open networks. The prominent example of eavesdropping is not covered, as anyone is expected to

[4] See e.g. the infamous OpenSSL bug in Debian (CVE-2008-0166) or the Postfix Flaw from August 2008 (CVE-2008-2936).

[5] Access is only given by means of an encrypted and authenticating VPN.

have heard about this already. Instead possible issues include how to detect computers which are eavesdropping, or how to avoid being detected, if one self is doing it. Also, how to detect software running on a specific port, even in case it delivers no or a fake banner is regularly part of the lecture. Further issues with networking are illustrated with man-in-the-middle attacks on rather unusual protocols like e.g. NTP.

Cryptography is only shortly touched. The main reason is that it virtually never is the weakest point in a IT system, therefore attacking cryptography makes no sense for reasonable attackers. However, the students are informed about side channel attacks on e.g. timing, or the effect of weak and predictable random number generators.

Web applications are ubiquitous today – and with the advent of powerful scripting languages like PHP and Python a lot of programmers try their luck. On the downside, this results in about 97% the web applications having security problems[6]. Dominant topics in this area is the abuse potential of Javascript and AJAX and (blind) SQL-injections, remote file inclusion, and Google hacking are covered.

Binary exploitation has been a major topic in the early years, but is currently on the decline. There are two reasons for this: the first is similar to not covering a lot of cryptography, i.e. binaries are getting much harder to exploit; as there are usually easier ways into a system, they do not form an easy attack vector anymore. Secondly, while there are still ways to cope with the increasing difficulties, the required knowledge about file formats, memory layouts, processor peculiarities, and more is getting so high that putting this into the limited time frame available does not fit. Still, overflows in various memory regions are discussed, as well as the effect of issues with unchecked integers (signed/unsigned, overflow), and format strings. As a last resort, the concepts of fuzzying, and some reverse engineering techniques can be introduced for completeness reasons.

Additional insights can be given from a selected set of sub topics, chosen based on the students choice: forensics and physical security are two of them.

The annual highlight is the *social engineering* challenge. An unaware, humorous and tolerant, target person is chosen, which is have to be well known to the organisers of the lecture. Then, the students receive the task to approach the target person on any topic of their will, and pretend to be the teacher from the course of the RWTH. The task is solved successfully, if they receive a response from which it is obvious that the target person did not notice that the original request was not send from the person it pretended to be. A debriefing of this event is obligatory. Every course is also given legal and ethical advise.

2.2 Ruhr-University Bochum

The Ruhr-University Bochum started to teach offensive IT-Security in 2007. Primarily the course *hacking lab* is a group based assignment which focuses on a different offensive IT-Security subject for every instance of the course.

[6] See e.g. http://www.webappsec.org/projects/statistics/

The project that has to be solved by the participating students is split into parts allowing the students to gather knowledge while already developing solutions to the prevoiusly assigned parts. The groupwork is one of the crucial elements of the course. On the one side it increases motivation for the students and on the other side helps them to develop strong communication skills most important for real world offensive IT-Security.

Offensive IT-Security as teached at the Ruhr-University tries to combine multiple views to the actual underlying problem. While the course participants are instructed to build a secure software for attack and defence on one hand, they are instructed by the lectures how to still find ways to circumvent the applied defensive mechanisms on the other hand. The aquired knowledge is then used by the students to attack the software developed by the other teams.

This scenario leads to a competitive situation between the teams which reflects the real world in small scale giving the lecturer the possibility to have the teams think about what has been achieved and how.

The software projects that have been covered so far are *nix daemons with various simple network services and win32 worms with a centralized management structure. While the projects differ quite a lot in the initial project definition the general method of knowledge transfer stays the same.

Tasks presented to the students, are wrapped into a role playing game reflecting the scenario of the assignment. For the *nix daemon development the participants had to impersonate the roles of developers and security researchers. While the role of the developer was to code safely and conservative the security reseachers were instructed to think of creative ways to get around the safety measures. In the case of the win32 worm development small virtual crime syndicates with different roles were assigned to the students. The distinct roles that have been given to the students where modeled to mirror the revenue generating process on which syndicates rely to the different parts of the development.

Therefore both of the scenarios enabled a vital discussion between the students and the lecturers about the technical and ethical aspects of the project assignment.

Lectures in the Ruhr-University course were held by IT-Security professionals invited to talk about a specific topic in the context of the assignment given to the participants. The possibility to not only gather knowledge about the presented work but also have an insight on real life experiences closely linked to the work in progress and discuss problems and ideas with the professionals gave the students another advantage over a regular IT-Security class.

3 Ethical Discussion

A common argument against the proposed solution is that teaching student offensive techniques, i.e. how to attack and compromise IT systems, is dangerous. In this section we will try to explain, why we think, there are good reasons to teach attacking IT systems. As we consider this an open discussion on *ethical*

issues, we neither expect nor give a final answer. In case the gentle reader is of different opinion we're open for discussion[7].

Before we get into any details, and in order to thwart arguments that our discussion is too biased, we like to point out that we know that there actually is a non-negligible potential of abusing the material presented in our lectures.

As it is the case with a common ethical discussion, there are two points to ponder: are the benefits (creating an open mind) worth the risk (abuse which may happen as a result of the course)? What are the benefits, and what are the risks? Obviously, the possible benefits are impossible to be quantified by any conventional metric. This also refers to the taught lessons on IT security. On the other hand, the possible risks also include a number of scenarios, where a possible damage can not be quantified; examples include leakage of private data, unauthorized use of other peoples' IT resources or in extreme cases: bodily harm to people in case of critical infrastructure failures.

In order to gradually build up some arguments lets start with the targets and (perceived) benefits of our approach. The main goal of our undertakings is to teach students an open mindset. Positive side effects are that the students learn some theoretical and a lot of practical information on IT security issues. Minor positive results are that the students can join exciting lab sessions and enjoy themselves.

In our point of view, training hacking is a good method of achieving the declared goals. This is backed up by a broad public opinion that the skill of creativity, the ability to find unconventional solutions, and the willingness to trace down a question to its very roots are positive and desirable properties. Related views on the world are e.g. *freethinking*.

Some advocates of teaching offensive IT security claim additional gains: allegedly, students who learn offensive techniques are better prepared to defend systems in real situations. Even though this claim still waits for empirical proof, it seems sound[8].

This brings us to the question, what kind of malice can originate from these courses. To a certain extend it is likely that participants may sooner or later come across the situation there they by chance encounter an obvious security flaw which is trivial to exploit. Even if they decide for exploitation in single cases, the damage inflicted will vary from nothing to the potential crash of a system. However, the participants may also choose to disclose the security vulnerability to the respective authorities or administrators, which in turn are able to remove the issue before any harm can be done by third parties.

Another question is, how likely it is that a participant may actively start to use his (or her) knowledge for malicious behavior, like e.g. writing malware, start phishing, or commit online fraud. If this would happen, then of course there would be damage, as these are criminal acts. We can see that there are multiple

[7] Feel free to send us your opinion.

[8] Roger Johnston claims that "knowing the vulnerabilities" is better than "to focus on the threat" http://www.cl.cam.ac.uk/~rja14/musicfiles/preprints/Johnston/securitymaxims.ppt hence backing up our argument.

future scenarios, some of them having positive effects, others not. Can we *trust* the students to not abuse their knowledge? Is the dark side of the power really more powerful? Can students be tricked into abusing their skills by fraudsters against their will?

Basically, trust is a prediction of reliance on an action, based on what one party knows about another party[9].

One possible way of evaluating this situation is with the help of Kohlberg's stages of moral development[10]. Typical adults, and other people who reason in a conventional way are usually considered to have reached the "fourth stage", i.e. for them it is important to obey laws and social conventions. We find it important to point out that at this stage, adults have the insight of adhering to social conventions is necessary because they know that adherence is a unconditional pre-requisite for any functional society. This is opposed to the lower stages where adherence is merely based on "following the law because the society asks you to do so".

But we can do better: Kohlberg states that about one fourth of humans reaches the fifth stage. Even accepting the risk to be foolhardy, we argument that students in general can be considered intellectually better off. Therefore there is a good chance that most of the students, especially if shown proper guidance, will soon reach the fifth stage. At this stage, students will accept thoughts on justice and social utility as a base for their actions. In some ethical dilemmas, the terms of *negative* and *positive rights* are used to discuss possible ethical implications. Positive rights are those rights which permit or oblige action, whereas negative rights are those which permit or oblige inaction. Typically, it takes more to justify an interference than to justify the withholding of goods or services. If we follow these lines of arguments, the possible interference with other peoples' goods by the students would demand the withholding of the benefits for the students. Obviously, this goes along with the traditional point of view in security courses.

However, in our case the negative results emerge only with a certain (unknown) probability. As we have seen in the previous discussion, it is legit to estimate this probability as "low", or even "very low". We can assume that one share of the students will use this knowledge for helpful purposes of even more people, e.g. by protecting networks, developing secure software or even carry on teaching.

Again, we face the problem that there is no quantification for neither human advancements, nor IT security – even though the later seems trivial compared to the first. This makes *utilitarian* ethical considerations impossible by any standard. If we reflect the problem from a *consequentialist* point of view, we see that any kind of malice is wrong, but foreseeable positive consequences make some risks acceptable.

A *virtue ethicist*, however, would consider that the main goal is to enhance the student's mindstate. This is, according to e.g. Aristotle, the proper goal of human life itself, as we exercise *reasoning*. This includes that a bearer of this

[9] http://en.wikipedia.org/wiki/Trust_(sociology)
[10] http://en.wikipedia.org/wiki/Kohlberg%27s_stages_of_moral_development

skill also possesses the virtues to oversee possible consequences of his actions. This goes hand in hand with Kohlberg's analysis.

Finally, one of the central sentences of Kant is "sapere aude"[11], an attitude which very closely resembles the mind set, we want to teach.

3.1 Practical Issues for IT security

In addition to the ethical discussion, we would like to point out a second reason, why teaching hacking is advantageous: Karl Popper's idea of *falsifiability* means that an assertion can be shown false by an observation or an experiment, but it is not possible to *verify* a hypothesis.

Our *conjecture* is that IT security for real computer systems is highly likely to be a topic where falsifiability is actually the only way to proof assumptions about security; i.e. it is impossible to proof that a computer system of non-negligible size is secure.

Arguments in our favour are that the mere complexity of modern computer systems, e.g., typical end-user systems, even more networks of computers, is too large to take all input variables into account which influence the system's operation. Examples of highly complex variables are the size of the code base of modern operating systems and applications, electromagnetic irradiation of screens and cables, the complexity of modern hardware and finally any human designer, programmer, operator or user of the system.

Despite the fact that there are formal proofs for the correctness, security and safety of code, all of these proofs necessarily need to make simplifying assumptions at least about hardware, physical laws and humans. This implies that any of these assumptions can be invalidated under certain conditions, rendering the proof insignificant for real world scenarios, i.e. it can not be assumed to hold under *every possible* situation in a real world.

Finally, the fact that compound IT systems do not necessarily take over security property of their components implies that compartmentalisation by itself can not help to break down the problem into smaller pieces. Hence, in the absence of (positive) means to proof the security of a system, the knowledge and art of attacking systems as a way to proof their vulnerability becomes the only feasible and scientific way to work in the area of IT security.

4 Conclusion

Taking these considerations into account, it is our opinion that the benefits are worth the risks. We believe that students, given advice and help, will themselves be able to fully understand the impact of their actions; not only with regards to legal issues, but also and especially on ethical aspects. This does not mean that slips are completely out of scope, but that the positive aspects will make up for them.

[11] Originally based on a phrase by Horace.