Experiences from Educating Practitioners in Vulnerability Analysis

Stefan Lindskog, Hans Hedbom, Leonardo A. Martucci, and Simone Fischer-Hübner

Department of Computer Science
Karlstad University, SE-651 88 Karlstad, Sweden
{stefan.lindskog|hans.hedbom|leonardo.martucci|simone.fischer-huebner}
@kau.se

Abstract. This paper presents a vulnerability analysis course especially developed for practitioners and experiences gained from it. The described course is a compact three days course initially aimed to educate practitioners in the process of finding security weaknesses in their own products. After giving an overview of the course, the paper presents results from two different types of course evaluations. One evaluation was done on-site at the last day of the course, while the other was made 3–18 months after the participants had finished the course. Conclusions drawn from it with regard to recommended content for vulnerability analysis courses for practitioners are also provided.

1 Introduction

The ongoing trend of a growing number of security vulnerabilities, threats and incidents has increased the efforts of IT industry to invest in the development of more secure systems. Vulnerability analysis (VA) is an important mean for improving security assurance of IT systems during test and integration phases. The approach that VA takes is to find weaknesses of the security of a system or parts of the system. These potential vulnerabilities are assessed through penetration testing to determine whether they could, in practice, be exploitable to compromise the security of the system. The Common Criteria have requirements on VA to be performed for the evaluation of systems with an Evaluation Assurance Level 2 (EAL2) or higher [2].

Upon request by a major IT company, our Department developed a compact VA course to be held on three working days for an international and heterogeneous, in terms of knowledge in the security area, group of practitioners from industry. Experiences and lessons learned from supervised student penetration testing experiments within an applied computer security course held at our Department [5] provided us with some inputs for the preparation of this VA course.

The emphasis of our VA course developed for industry was put on practical, hands-on experiments. The course outline and first experiences gained from the course held in 2005 were first presented in [6]. Meanwhile, after the course has been held at our department five times with an average number of 16 participants

in 2005 and 2006, we were interested in a more detailed evaluation by means of a statistical survey. Our aim was to investigate how useful the participants have perceived the practical experiments and course content for their jobs, what influence it has had on their work, and whether they think that the course has helped to improve the overall quality of the test procedures applied by their companies. The results of this survey and conclusions drawn from it with regard to the recommended content of compact VA courses for IT industry will be presented in this paper.

The rest of the paper is organized as follows. Section 2 gives a brief overview of the course and its content. Especially the hands-on assignments are presented. In Section 3, the on-site course evaluation is discussed, whereas Section 4 describe the results achieved from the post course evaluation performed. Experiences from the course and conclusions are provided in Section 5.

2 Course Overview

This section will give a brief description of the course. A more thorough discussion can be found in [6]. The requested course was initially aimed for software testers with no or little knowledge about security in general and VA in particular, but with an extensive knowledge in software testing. However, as will be discussed later, in reality the participants had a more diversified background and very few actually worked as testers. Since the target group was practitioners, a practical course was requested. Approximately 1/3 of the course cover theoretical aspects and 2/3 is used for practical hands-on assignments. The latter was intended to give the attendees hands-on experience on how to conduct a VA of either a software component or a complete system. A three days course (24 hours course) was selected as the best choice for the course length, since the participants should not be absent from their tasks for a long period. The course is divided into the four following blocks: (1) introduction to computer and network security, (2) computer and network security protocols and tools, (3) VA, and (4) known vulnerabilities, reconnaissance tools, and information gathering. The following 10 hands-on assignments are provided in the course:

1. **Password cracking with John the Ripper.** In this experiment, a local password cracker application is experienced. John the Ripper v.1.6, a password cracker tool, which is intended to detect weak UNIX passwords [11] is used. This tool is both easy to use and easy to deploy. Synthetic (artificially populated) `passwd` and `shadow` files from a Linux box is provided.
2. **Testing for randomness using the NIST test suite.** In this experiment, the NIST Statistical Test Suite [8] is used to evaluate outputs from different pseudo random number generators (PRNG). It implements 15 statistical tests. It has also embedded implementations of well-known bad PRNG, such as the XOR RNG, and also NIST approved RNG, such as the ANSI X9.31. Sample data files are also provided as companion part of this test suite.
3. **Network sniffing using Ethereal.** The assignment is divided in two parts, each part with a different network topology. In the first part, the participants

verify that a popular network protocol, i.e., TELNET, is weak regarding security using a network analyzer tool [3]. In the second part, a rather insignificant change in the network topology is made that modified the test result and it is up to the participants to explain why the result changes.

4. **ARP spoofing using Cain & Abel.** In this assignment, the participants test an ARP spoofing tool [10] on a switched network, verify the achieved results, and explain the results accordingly.
5. **Black-box testing using the PROTOS tool.** In this experiment the PROTOS tool [12] is used as a black box testing tool against the SNMP protocol implementation of Cisco 1005 router running IOS 11.1(3). The PROTOS Test Suite c06-snmpv1 is here used to perform a Denial of Service (DoS) attack against the Cisco router.
6. **Firewall configuration.** This exercise is based on an assignment originally published in [7]. The participants are divided into groups of two. Each group is given a description of a setup and is asked to write firewall rules in Linux using `ipTables` implementing a given policy defined in the problem statement.
7. **Node hardening using Bastille**. Since the participants have a heterogeneous background regarding in-depth knowledge of different operating systems, the open source tool Bastille [1] is selected for the this assignment. The tool lets the user answer a number of questions on how they want the computer to be configured and then configures it according to the answers. The participants were asked to make their own computers as secure as possible given that they still should be functional as networked computers.
8. **Port scanning with NMAP.** The goal of this experiment is gathering information about two running systems. Two workstations configured as servers (one Linux and one Windows server) are used as target systems. The participants run the Network Mapper (NMAP) [4], an open source port scanning tool, under Linux. Servers run several network services, such as FTP, HTTP, NetBIOS, etc. The participants have to find and identify the servers in a given IP network range, since their IP addresses are not provided, find out the operating system running, and identify the open ports in each server
9. **Security scanning with NESSUS.** In this exercise, two target servers are set up. One Windows 2000 (set up as a domain server) and one running Fedora Core 3 Linux. None of the servers are patched, running all standard services and acting as target systems for the security scanner [9]. The participants are divided into pairs and they are asked to find the servers IP addresses and to find out which operating systems they are running. After finding the servers, the participants had to scan them and report all the vulnerabilities found.
10. **Final assignment (putting it all together).** The last assignment is a full-day final practical assignment that concludes the course, a "putting it all together" experiment that summarizes the full VA process. In this exercise, the participants are divided into groups of four. Each group is given a Fedora Core 3 Linux server with all services running. Every group also got

a requirement document describing the role of the server and the security requirements on it. They are asked to find out what needed to be done in order to fulfill the requirements and also to perform the changes and verify the results. In order to do this they had access to all the tools used in the previous exercises and a list of useful Internet links. They also had access to the Internet and are told that they could use it freely. Moreover, they may also use any other freely available tool found on Internet. The participants had to report all the miss-configured parameters, vulnerabilities and also had to suggest changes in the system in order to adhere to the specification. The results were discussed in a summing-up session just after the exercise.

3 On-Site Course Evaluation

In this section, we present the results from the on-site evaluation that was performed as the last activity within the course. The participants assessed the usefulness of the 10 assignments within the course. The questionnaires were answered individually and anonymously.

The main question regarding the evaluation of the assignments were formulated as follows: "*An important part of the course is the hands-on assignments. Please give your opinion about the usefulness of these using the 1–7 scale (1 means poor and 7 means excellent)*". The participants were encouraged to motivate their answer. Table 1 summarizes the result from the on-site evaluations. The results are presented in percentages of the total feedback for each assignment and the most likely grade, i.e., the statistical type value, is highlighted in boldface. The presented results were based on a total of 60 evaluations collected from the participants in five course instances from spring 2005 to autumn 2006.

Table 1: Results from the question: "Please give your opinion about the usefulness of the assignments using the 1–7 scale (1 means poor and 7 means excellent)".

Grade	**Assignments**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1				2%				2%		
2		16%			7%					3%
3		8%	5%	3%	7%	2%			2%	7%
4	11%	**24%**	10%	10%	25%	13%	10%	8%	3%	22%
5	39%	**24%**	27%	32%	25%	23%	22%	31%	23%	31%
6	**47%**	**24%**	**45%**	**42%**	**33%**	**47%**	**53%**	**47%**	**50%**	**32%**
7	3%	3%	13%	10%	3%	15%	15%	12%	22%	5%

The assignment considered the most useful by the participants was the security scanning using the NESSUS, followed by the port scanning assignment with NMAP. The least useful assignment, according to the on-site evaluation, was the test for randomness using the NIST tool. According to the feedback collected from the participants, the security and port scanning tools could be easily deployed and used during test phase, but testing for randomness was a

fairly more uncommon and also a fairly slow test in comparison to the other assignments.

All the participants that have taken the course so far have either been satisfied or very satisfied. Additional results from the on-site evaluation is given in [6].

4 Post Course Evaluation

The post course evaluation was conducted in January 2007, which was 3–18 months after the participants had taken the course. The evaluation was performed based on a web based questionnaire. The prior participants were contacted by email to voluntarily provide their input within one week. All in all, 55 emails were sent out to prior course participants and 22 of them filled in the questionnaire. The questionnaire contained the following 12 questions:

1. *Position when the course was given*
2. *Number of years in that position*
3. *Current position*
4. *Country of residence*
5. *Education (Technical degree, Management degree, or Other)*
6. *Did the course fulfill your expectation? (Yes or No)*
7. *What is the most important learning outcome from the course?*
8. *To what degree have the course helped you in your position? (1 means not at all and 7 means very well)*
9. *An important part of the course is the hands-on assignments. In retrospect, please give your opinion about these using the 1-7 scale. (1 means poor and 7 means excellent)*
10. *Do you practically apply any of the tools/techniques from the assignments in your current position? (Never, Sometimes, or Often)*
11. *Do you think that the knowledge gained from the course has helped to improve the overall quality of the test procedures you apply? (Yes, No, or I don't know)*
12. *Other comments/suggestions*

The indent with the first question was to investigate the respondents' job positions when they took the course. Based on the input five main job categories could be distinguished (number of persons in each categories are specified within parenthesis): software tester (5.5[1]), technical specialist (6) software engineer (3.5), system manager (4), and manager (3). To the manager category project, product, and platform managers are counted. The number of years in that position varied between 0 and 10 years. From question 3 we found that one software tester and one system engineer had advanced to a system manager position. The number of persons in the other two categories had not changed. From the country of residence question, we found that 2/3 of the respondents were from Sweden and the rest from abroad. These figures reflect the distribution of the participants that have taken the course. 20 out of 22, i.e., 91%, reported a technical degree and the other two in the other category. All 22 respondents answered that the course did fulfill their expectations. The most important learning

[1] One respondent reported a shared position as software tester and software engineer.

outcomes that were reported in the questionnaires are (1) knowledge about and experiences with the tools, and (2) awareness of security in general and VA in particular. The results from question 8 is illustrated in Fig. 1.

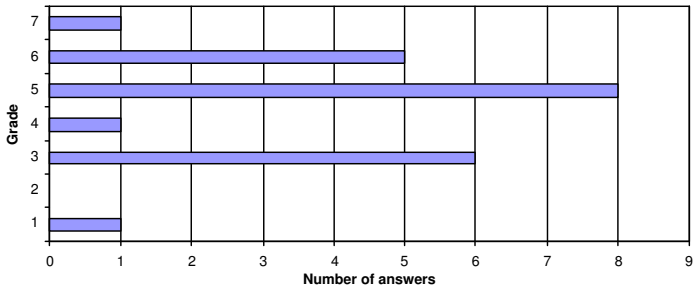

Fig. 1: Results from question 8: "To what degree have the course helped you in your position? (1 means not at all and 7 means very well)".

From the figure it is evident that more than 60% of the respondents reported that the course have helped them in their positions quite much, much, or very much (i.e., grade 5–7). The respondent that reported that the course has not at all helped was, at the time the course was taken, acting as a manager and is still in that position. Remember from Section 2 that the course was not targeted for that job category.

In question 9, the respondents were asked to give their opinions about the different assignments in retrospect. The result is presented in Table 2. The statistical type value is again highlighted in boldface.

Table 2: Results from question 9: "In retrospect, please give your opinion about the assignments using the 1-7 scale. (1 means poor and 7 means excellent)".

Grade	**Assignments**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1										
2	9%	23%			5%		9%		5%	
3	14%	5%		14%	5%			5%		9%
4	9%	**32%**	14%	14%	23%	18%	9%	23%	14%	18%
5	**32%**	18%	23%	**32%**	**32%**	32%	**41%**	**32%**	23%	18%
6	27%	23%	**50%**	**32%**	18%	**45%**	32%	23%	**41%**	**41%**
7	9%		14%	9%	18%	5%	9%	18%	18%	14%

From the table it is clear that the respondents were satisfied with all 10 assignments. The most popular assignments in retrospect were assignments 3 (network sniffing using Ethereal) and 6 (firewall configuration). Remember from Section 3 that these were not graded highest in the on-site evaluation. Instead, assignments 8 (port scanning with NMAP) and 9 (security scanning with NESSUS) were the most popular ones.

Question 10 was asked to investigate whether the participants have applied the tools/techniques used in the assignments professionally. The results are presented in Fig. 2. From the figure it is evident that knowledge gained from as-

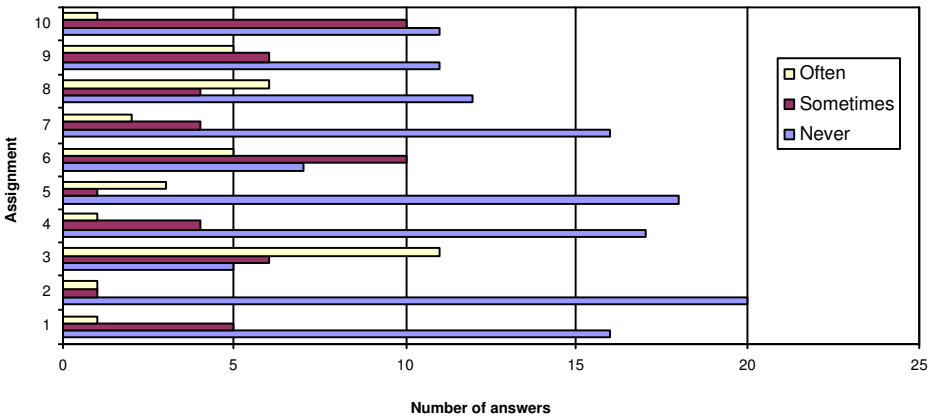

Fig. 2: Results from question 10: "Do you practically apply any of the tools/techniques from the assignments in your current position? (Never, Sometimes, or Often)".

signments 3 (Network sniffing using Ethereal) and 6 (Firewall configuration) are either used often or sometimes by 17 and 15 respondents, respectively. Hence, the tools used in these assignments are the ones that are used most frequently by the respondents after the course. This is probably also the reason why these assignments are graded highest in the previous question. Similarly, the least used tool reported by the respondents was the one introduced in assignment 2 (Testing for raddomness using the NIST test suite). This assignment was assigned the lowest grade of all 10 assignments in both the on-site and post evaluation.

In question 11, the respondents were asked whether they believe that the course has helped to improve the quality of the test procedures and 17 (i.e., more than 77%) answered "Yes". The remaining five respondents answered that they did not know.

5 Experiences and Conclusions

Our experience is that during a compact three days course in VA aimed for practitioners many different issues can be covered. The respondents from the post evaluation have reported that most of them are convinced that the knowledge gained from the course has helped them to produce products with higher quality than before. From the various instances of the course and through the on-site evaluations and the post evaluation, we have learned that the mixture of practical hands-on assignment during 2/3 of the time and lectures 1/3 of the time are suitable when educating practitioners in VA. We have also learned that the participants' opinion about assignments might change over time.

Assignment 2, i.e., testing for randomness using the NIST test suite, needs special attention. This assignment was added to the course on request from the contractor because of its importance for developing secure cryptographic systems. Based on the on-site evaluations and post evaluation very few of the participants have graded this assignment very high. In addition, only two of

the respondents in the post evaluation have reported that they have used the knowledge from this assignment afterwards. Hence, for future course instances we plan to skip this exercise, but still cover the theory on testing for randomness in the course.

From both the on-site evaluations and post evaluation it is clear that the course has been appreciated by the participants. The course has, furthermore, steadily been changed based on the participants' invaluable comments and suggestions for enhancement, but also based on the lecturers feelings from the five different course instances so far conducted. Except assignment 2, we believe that the theory as well as the assignments covered in the course together provides a very well balanced VA course for practitioners. We also believe that the course, or at least parts of it, would fit nicely in the third year of Bachelor programs in computer science, computer engineering, or information technology.

On request form the contractor, and some of the course participants, a one day follow-up course is now under development. The follow-up course will focus on: wireless networks, authentication, authorization, and accounting (AAA) architectures, and virtual private networks (VPNs).

References

1. Bastille Linux. The Bastille hardening program: Increased security for your OS. http://www.bastille-linux.org/, Accessed January 23, 2007.
2. Common Criteria Implementation Board. Common criteria for information technology security evaluation, version 3.1. http://www.commoncriteriaportal.org/, September 2006.
3. Ethereal, Inc. Ethereal: A network protocol analyzer. http://www.ethereal.com, Accessed January 23, 2007.
4. Insecure.org. Network mapper. http://insecure.org/nmap/, Accessed January 23, 2007.
5. S. Lindskog, U. Lindqvist, and E. Jonsson. IT security research and education in synergy. In *Proceedings of the 1st World Conference in Information Security Education (WISE'1)*, pages 147–162, Stockholm, Sweden, June 17–19, 1999.
6. L. A. Martucci, H. Hedbom, S. Lindskog, and S. Fischer-Hübner. Educating system testers in vulnerability analysis: Laboratory development and deployment. In *Proceedings of the Seventh Workshop on Education in Computer Security (WECS'7)*, pages 51–65, Monterey, CA, USA, January 4–6, 2006.
7. Mixter. Gut behütet. *C'T – Magazin für Computer Technik*, pages 202–207, June 17–19, 2002.
8. National Institute of Standards and Technology (NIST). NIST statistical test suite. http://csrc.nist.gov/rng/rng2.html, Accessed January 23, 2007.
9. Nessus Project. Nessus vulnerability scanner. http://www.nessus.org/, Accessed January 23, 2007.
10. Oxid.it. Cain & Abel. http://www.oxid.it/, Accessed January 23, 2007.
11. Openwall Project. John the ripper password cracker. http://www.openwall.com/john/, Accessed January 23, 2007.
12. University of Oulo. PROTOS security testing of protocol implementations. http://www.ee.oulu.fi/research/ouspg/protos/index.html, Accessed January 23, 2007.