Guidelines: Which SEED Labs Should I Use?

Over the last few years, we have worked with a number of instructors who adopted our labs in their classes. Based on the experience, we have summarized the following guidelines to help you decide whether to adopt our labs or not, how to adopt our labs, what labs to adopt, etc.

1 Overview

Deciding what labs to adopt is up to you and is dependent on how you would like to teach a security course. Before we started this project, we have conducted a survey to investigate how the instructors at different universities teach computer security courses. The results indicate that, unlike some other courses such as Operating System and Networking, there are no well-adopted consensus on what should be covered in computer security courses. Giving the unique nature of computer security, it might be a wrong direction to develop such a consensus.

Having in mind the wide variety of ways of teaching computer security, we develop our labs not based on a specific syllabus, but instead, we want our labs to cover as many aspects of computer security as possible; this way, instructors can always find some labs that are suitable for their security courses regardless of how they teach the courses. To help instructors find out what labs are appropriate for their courses without knowing how they teach the courses is not easy; however, we have developed the following two strategies that can help achieve this goal: the *principle-based strategy* and the *course-based strategy*. In the principle-based strategy, we categorize our labs based on the principles of computer security; in the course-based strategy, we categorize our labs based on some specific courses.

We have also mapped our labs to the chapters of several popular textbooks that are widely used by computer security instructors. This mapping will help instructors decide what labs they can use if they are using a specific textbook.

2 Mapping SEED Labs to Security Principles

Regardless of how instructors teach computer security and in what contexts (e.g. networking, operating system, etc.) they teach computer security, one thing is for sure: they should cover the principles of computer security. In civil engineering, when building bridges, there are well-established principles that need to be followed. Security engineering is no difference: in order to build a software system that is intended to be secure, we also need to follow principles. Regardless of how computer security is taught, the fundamental principles that most instructors cover are quite the same, even though the principles might be covered in different contexts.

The definition of "security principles" is interpreted differently by different people: some interprets it as software engineering principles, such as principle of least privileges; some interprets it as access control, authentication, etc. To avoid confusion, we use the following definition:

> *A computer security principle is an accepted or professed rule of action or conduct in building a software or hardware system that is intended to be secure.*

We have categorized our labs based on the fundamental computer security principles, including Authentication (AU), Access Control (AC), Cryptography (CG), Secure Programming (SP), and Secure Design (SD). The categorization is described in Table 1.

As for the types of labs, we divide the SEED labs into three categories based on the intentions of these labs. Each type of labs requires different skills and may need different amount of time to finish:

- *Vulnerability and Attack Labs:* The goal of these labs is to achieve learning from mistakes. Vulnerabilities are often caused by mistakes in design, implementation, and configuration. These labs give students the opportunity to have hands-on experience with real vulnerabilities. In these labs, students need to identify vulnerabilities, develop attacks to exploit vulnerabilities, fix the vulnerabilities, and defend against the attacks.

- *Design and Implementation Labs:* The goal of these labs is to achieve learning by system development. They allow student to apply security principles, concepts, and ideas to build a secure systems in a lab environment.

- *Exploration Labs:* The goal of these labs is to achieve learning by exploring. They permit students to explore an existing system to understand the intended security principles, concepts, and ideas. Exploration labs are like a "guided tour" of a system, in which, students can "touch" and "interact with" the key components of a security system to learn the principles of security.

3 Mapping SEED Labs to Security Courses

After studying a number of security courses taught at different universities and colleges, we have identified several representative types of courses, and made suggestions regarding what SEED labs are appropriate for these courses (Table 2).

1. *System-focused Courses:* This type of course focuses on security principles and techniques in building software system. Network, also considered as a system, might be part of the course, but not as the focus. The focus is mainly on software system in general. Operating systems, programs, and web applications are usually used as the examples in the courses.

   If an instructor wants to ask students to design and implement a real system related to system security, there are several choices. (a) If the instructor wants to let students learn how to use cryptography in a real system, the Encrypted File System Lab is a good choice. (2) If the instructor wants to let students gain more insights on access control mechanisms, the Role-Based Access Control Lab and Capability Lab are good candidates. (3) If the instructor wants students to learn some of the interesting ideas in improving system security, the Address Space Layout Randomization Lab and the Set-RandomUID Sandbox Lab are good candidates. All these labs are carried out in the `Minix` operating system because of the need to modify operating systems. These labs can be used as the final projects.

2. *Networking-focused Courses:* This type of course focuses mainly on the security principles and techniques in networking.

3. *Programming-focused Courses:* The goal of this type of course is to teach students the secure programming principles when implementing a software system. Most instructors will cover a variety of software vulnerabilities in the course.

4. *Software-Engineering-focused Courses:* This type of course focus on the software engineering principles for building secure software systems. For this type of courses, all the vulnerabilities labs can be used to demonstrate how flaws in the design and implementation can lead to security breaches. Moreover, to give students an opportunity to apply the software engineering principles that they have

learned from the class, it is better to ask students to build a reasonably sophisticated system, from designing, implementation, to testing. Our design/implementation labs can be used for this purpose.

4 Mapping SEED Labs to Popular Textbooks

Although independently developed, the SEED labs work well with most textbooks. To help instructors decide which labs to use, we have mapped our SEED labs to the chapters of several security textbooks. The mappings, summarized in Table 3, are conducted on the following six textbooks. As new textbooks, new editions, and new labs become available in the future, we will update the table accordingly.

- *Introduction to Computer Security (2004)*, by Matt Bishop. We refer to this book as *MB1*.
- *Computer Security: Art and Science (2002)*, by Matt Bishop. We refer to this book as *MB2*.
- *Security in Computing (3rd Edition, 2003)*, by Charles P. Pfleeger and Shari Lawrence Pfleeger. We refer to this book as *PP*.
- *Network Security: Private Communication in a Public World (2nd Edition, 2002)*, by Charlie Kaufman, Radia Perlman, and Mike Speciner. We refer to this book as *KPS*.
- *Introduction to Computer Security (2011)*, by Michael T. Goodrich & Roberto Tamassia. We refer to this book as *GT*. The book acknowledges the SEED project; its web site (`http://www.securitybook.net/`) provides direct links to thirteen of the SEED labs.
- *Computer Security (3rd edition, 2011)*, by Dieter Gollmann. We refer to this book as *DG*.

Types	Labs	AU	AC	CG	SP	SD
Vul. & Attack Labs	Buffer-overflow Vulnerability Lab				UG	
	Return-to-libc Attack Lab				UG	
	Race-Condition Vulnerability Lab				UG	
	Format-String Vulnerability Lab				UG	
	`Chroot` Sandbox Vulnerability Lab		UG			UG
	`Set-UID` Program Vulnerability Lab		UG			
	TCP/IP Attack Lab				UG	UG
	DNS Pharming Attack Lab					UG
	Cross-Site Scripting (XSS) Attack Lab		UG		UG	UG
	Cross-Site Request Forgery (CSRF) Attack Lab		UG			UG
	SQL Injection Attack Lab		UG		UG	UG
	ClickJacking Attack Lab		UG			UG
Exploration Labs	`Linux` Capability Exploration Lab		UG			
	Web Browser Access Control Lab		UG			UG
	Packet Sniffing & Spoofing Lab		UG			UG
	Secret-Key Crypto Lab			UG		
	One-Way Hash Function Crypto Lab			UG		
	Public-Key Crypto Lab	UG		UG		
	Pluggable Authentication Modules Lab	UG				UG
	SYN-Cookie Lab			UG		
Design Labs	`Linux` Virtual Private Network (VPN) Lab	G	G	G	G	G
	`Linux` Firewall Lab		G			
	`Minix` Firewall Lab		G			
	`Minix` IPSec Lab	G	G	G		G
	`Minix` Capability Lab		G			G
	`Minix` Role-Based Access Control Lab	G	G		G	G
	`Minix` Encrypted File System Lab	G		G	G	G
	`Minix` Set-RandomUID Sandbox Lab		G			

Computer Security Principles:
AU = Authentication, **AC** = Access Control, **CG** = Cryptography,
SP = Secure Programming, **SE** = Secure Design.

Table 1: Principle-Based Classification of SEED Labs ("UG" indicates that this lab is appropriate for both undergraduate students and graduate students, "G" indicates that the lab is appropriate for Graduate students only, and not appropriate for average undergraduate students.)

Types	Labs	Weeks	System	Network	Prog.	SE
Vul. & Attack Labs	Buffer-overflow Vulnerability Lab	1	UG	UG	UG	UG
	Return-to-libc Attack Lab	1	UG	UG	UG	UG
	Race-Condition Vulnerability Lab	1	UG		UG	UG
	Format-String Vulnerability Lab	1	UG		UG	UG
	`Chroot` Sandbox Vulnerability Lab	1	UG			UG
	`Set-UID` Program Vulnerability Lab	2	UG		UG	
	TCP/IP Attack Lab	2		UG	UG	UG
	DNS Pharming Attack Lab	1		UG		UG
	Cross-Site Scripting (XSS) Attack Lab	1		UG	UG	UG
	Cross-Site Request Forgery (CSRF) Attack Lab	1		UG	UG	UG
	SQL Injection Attack Lab	1		UG	UG	UG
	ClickJacking Attack Lab	1		UG		UG
Exploration Labs	`Linux` Capability Exploration Lab	2	UG			
	Web Browser Access Control Lab	1	UG	UG		
	Packet Sniffing & Spoofing Lab	1		UG		UG
	Secret-Key Crypto Lab	1	UG	UG		
	One-Way Hash Function Crypto Lab	1	UG	UG		
	Public-Key Crypto Lab	1	UG	UG		
	Pluggable Authentication Modules Lab	1	UG			UG
	SYN-Cookie Lab	1		UG		
Design Labs	`Linux` Virtual Private Network (VPN) Lab	4		G		G
	`Linux` Firewall Lab	2		G		G
	`Minix` Firewall Lab	2		G		G
	`Minix` IPSec Lab	5		G		G
	`Minix` Capability Lab	4	G			G
	`Minix` Role-Based Access Control Lab	5	G			G
	`Minix` Encrypted File System Lab	5	G			G
	`Minix` Set-RandomUID Sandbox Lab	2	G			G

Table 2: Course-Based Classification of SEED Labs ("SE" stands for Software Engineering, "Prog." stands for Programming. The meanings of 'UG' and 'G' are the same as those in Table 1)

Types	Labs	MB1	MB2	PP	KPS	GT	DG
Vulnerability and Attack Labs	Buffer-Overflow Lab	20, 26	23, 29	3	-	3	10
	Return-to-libc Lab	20, 26	23, 29	3	-	3	10
	Race-Condition Lab	20, 26	23, 29	3	-	3	10
	Format-String Lab	20, 26	23, 29	3	-	3	10
	Chroot Sandbox Lab	20, 26	23, 29	3	-	3	-
	TCP/IP Attack Lab	20, 23, 26	23, 26, 29	3	-	5	17
	DNS Pharming Attack Lab	20, 23, 26	23, 26, 29	3	-	6	17
	Cross-Site Scripting Attack Lab	20, 23, 26	23, 26, 29	3	25	7	18
	Cross-Site Request Forgery Attack Lab	20, 23, 26	23, 26, 29	3	25	7	18
	ClickJacking Attack Lab	20, 23, 26	23, 26, 29	3	25	7	18
	SQL Injection Attack Lab	20, 23, 26	23, 26, 29	3, 6	-	7	10, 18
	`Set-UID` Program Vulnerability Lab	14	15	4	-	3	7
Exploration Labs	Pluggable Authentication Modules Lab	11	12	4	9, 10	3	4, 7
	`Linux` Capability Exploration Lab	12, 14, 17	13, 15, 19	4	-	3	5
	Secret-key Encryption Lab	8-10	9, 10, 11	2, 12	2-6	8	14
	One-Way Hash Function Lab	8-10	9, 10, 11	2, 12	2-6	8	14
	Public-key Cryptography Lab	8-10	9, 10, 11	2,12	2-6	8	15
	SYN-Cookie Lab	23	26	2, 7	5	5	17
	Packet Sniffing and Spoofing Lab	23	26	2, 7	5	5	17
	Web Access Control Lab	4, 14	4, 15	4, 7	25	7	18
Design and Implementation Labs	Set-RandomUID Sandbox Lab	19	22	-	-	3	-
	`Minix` Capability Lab	12, 14, 17	13, 15, 19	4	-	3	5
	`Minix` Role-Based Access Control Lab	12, 14, 17	13, 15, 19	4	-	9	5
	Encrypted File System Lab	8-10, 17	9-11, 13, 19	2, 4	2-5	9	14
	Address-space Layout Randomization Lab	22, 24, 26	25, 27, 29	4, 5	-	-	-
	IP Sec Lab	8-10, 17, 23	9-11, 19, 26	2, 7	2-5, 17	6	16
	VPN Lab	8-10, 17, 23	9-11, 19, 26	2, 7	2-5, 17	6	16
	`Linux` Firewall Lab	17, 23	19, 26	7.4	23	6	17
	`Minix` Firewall Lab	17, 23	19, 26	7.4	23	6	17

Table 3: The SEED Labs and Their Mappings to Textbooks. The numbers in the table are chapter numbers.