Laboratory on Offensive Computer Security

2IC80

Course guide

I. General information

Study year:	2018-2019; quarter 3
Contact hours:	After class
Responsible lecturer:	Dr. Luca Allodi (WIN, SEC)
Co-lecturers:	Mr. Guillaume Dupont (WIN, SEC)
Information:	l.allodi@tue.nl
Teaching schedule:	Lectures: 2x2 hours/week during 8 weeks
Studio classroom:	2x2 hours/week during 8 weeks
Examination:	Project (50%) + Written exam (2 hrs) (50%)

What this course is. This is an introductory course on hacking, with a focus on computer networks and software. The full course will revolve around the idea that (most) mechanisms can be illegitimately misused to obtain (some level of) desired behaviour; this is the very essence of computer security. We will explore network protocols and the implied trust relations they operate upon; we will look at software vulnerabilities and at how they can emerge from software code; we will look at exploit testing and automated attack frameworks; we will look at penetration testing and advanced networked attacks.

What this course is not. This course is **not** a condensed, spoon-fed manual for hacking: if there's any truth in the saying "*Give a man a fish and you'll feed him for one day; teach a man how to fish and you'll feed him for a lifetime*", this certainly is perfectly reflected in computer security at large: There is no useful "manual" for hacking, no prescribed rule or procedure one has to follow to perform a successful attack. Hence, this course does not pretend to be one.

What you can expect. Attending this course you will gain a both theoretical and practical understanding of the principles of hacking. At the end of the course you will have a basic understanding of the composition of (networked) computer systems, and understand how to exploit these constructs to make them perform actions they were not designed for. You will have gained the ability to critically think about system (in the general sense) architectures, and formalize (in code) ways around their core mechanics to obtain a not-predefined behaviour (e.g. to exfiltrate data or intercept network connections in LANs).

How this is going to happen. Each class is made of 2x2 hours slots. The first slot is dedicated to theory. The second is a **self-guided** lab activity with the *assistance* of the lecturers. **A laptop is required for the execution of the laboratory activities.**

Study material: All theory is covered in the slides and in additional study material provided during the course. The demo lab development (see *Examination*, below) will need additional material that the student is expected to look up during the lab development itself.

Prior knowledge: No hard requirement, but basic skills in programming (Python) and a basic understanding of computer networks and systems help. All basics will be covered during the course as a quick refresh.

II. Learning goals

The student will learn:

- the characteristics of software/architectural vulnerabilities and their impact on system security;
- malware types, functionalities, and propagation mechanisms;
- vectors for attack delivery; the different types of network and host defences and their limitations;
- how to engineer a working attack

III. Content

- Security of Network protocols – IP, TCP/IP, Application layer
- Network reconnaissance and mapping
- Vulnerabilities & attack surfaces
- Penetration testing
- Social engineering
- Attacks - web attacks
- Attacks - malware
- Attacks - advanced web attacks
- Privacy in networks

IV. Examination

Project (50%)
All students are expected to produce a laboratory demo choosing one of two options:

- reproducing *in code* **two** automated attacks among those implemented in the tool *Ettercap;*

- propose their own project idea, to be discussed with the lecturer.

All students will work in groups of 2 people, exceptionally of three (+1 attack to reproduce). Students that so wish may decide to work alone.

The examination is a short (10') presentation by groups (with **all** group members) where each member will be asked questions about the implementation of specific functionalities of the tool. Presentations will be scheduled ahead of time.

Written examination (50%)
The final examination will consist of open questions on the course material. No code/algorithms/math is involved in the final examination. The goal there is to show that you have a clear-enough understanding of attack mechanisms and root causes to explain them schematically or in plain English (as you'd have to do with a customer, were you a security professional).

V. Details on labs & project

Class laboratories

These labs happen after the first two hours of class. Students will need their own laptop. Group work is encouraged (e.g. 2 students using 1 laptop). All material needed for the lab activities is provided by the instructors.

The labs will consist in implementations or analyses of attacks discussed in class. All activities will be performed on one or more **virtual machines**, virtualized on VirtualBox (https://www.virtualbox.org). Guidelines on how to deploy the VMs will be published before the start of the course.

All students participating in the lab **must have already setup the VMs and be able to run them for the lab activity on day 1.**

Project

The lab demo is part of the final examination grade. It weights for 50% of the final vote.

The goal of the laboratory is to allow the student to play with the theoretical instruments provided in class in conjunction with the tools (e.g. to monitor network activities) used in the laboratory parts in order to **engineer** a new attack moulded on top of existing automated implementations.

In practice:

1. Form groups of 2 people. There will be an online shared document to do this. Students that wish to work alone may decide to do so. Students that do not have a group but wish to have one will be assigned one. Exceptional cases admit groups of 3

2. Decide which two attacks you intend to reproduce. You can start off from the tool Ettercap (https://ettercap.github.io/ettercap), or you can choose any other attack you might be interested in. Ettercap is convenient because:
    a. Network oriented
    b. Plenty of online documentation
    c. Activity can be easily monitored
    d. Plenty of plugins/add-ons to choose from
3. The baseline set of attacks is: automated, persistent ARP poisoning + DNS Spoofing
4. Perform the chosen attack(s) using the tool you chose (e.g. Ettercap)
5. Analyse the traffic generated during the attack. Key aspects:
    a. Understand what is the attack supposed to do;
    b. Map that understanding to the traffic that is generated by the tool;
6. Reproduce the key aspects of that behaviour in code (e.g. using scapy: http://secdev.org/projects/scapy/);
7. Your deliverable will be a technical report (max 10 pages) and a video (max 3 minutes) showing the tool in action.

VI. Frequently asked questions

Q. Am I supposed to attend all classes?

Of course yes; yet, this is a University course and you are very free to decide for yourself.

All theoretical aspects will be covered in class and the provided material is sufficient for self-studying as far as the written examination goes. Taking notes in class is always a (significant) advantage. The laboratory activities are very recommended as they will help you during the written exam by providing hands-on practice on the theoretical part.

Q. What does it mean that the laboratory is self-guided?

It means that the instructors will *not* perform the exercises on the projector or guide you through the slides during the lab hours, with a few exceptions only. You will execute the lab by yourself (or in small groups, 2/3 people max).

The reason why this is so is that part of the goal of this course is for the student to learn how to **critically investigate specific attack scenarios** by themselves**.** This **requires time and effort, and cannot be achieved by "copy-pasting" precooked solutions.**

All activities are however **guided** through a slide deck giving suggestions, pointing out specific technical steps to perform, and so on. Instructors will remain available to help you with specific aspects or questions you might have.

The material covered during that day's class is always *very* relevant to the laboratory activity. This is so that you can immediately apply in practice what we have covered in class.

Q. ***I do not know how to program!***

This is not an issue per se. However this is *not* a programming course, so it will *not* cover the principles of programming. But there is no hard pre-requisite either. The programming skills required are fairly low and can be quickly acquired by any wilful student that likes a challenge.

So the real question is: ***do you like a challenge***? If not, wrong course.
But I am sure you do.

Q. *Can I redo my lab project if I do not like the final score?*

No.

Q. *Are you available for technical suggestions/feedback for the demo development?*

There will be lab sessions dedicated exclusively to that. Students that show up can use that time to ask questions and review problems together with the instructor(s).

Appendix A. Course schedule

CLASS: Class; LAB: laboratory activity; FP: free practice; PRJFB: project feedback.

Module	Topic	Date	Hrs	type	Week	Lecture	Agenda	Content	Prj
Network layers	Intro and Networks	06/02/2019	2	CLASS	1	L1	Intro to course + security foundations	general info: exam, course content, lab activities; principles of (in)security	
		06/02/2019	2	LAB	1		Instruction	Hands on Linux, terminal	groups negotiations
		08/02/2019	2	CLASS	1	L2	Security of Network procols - IP	review of IP and attacks	
		08/02/2019	2	LAB	1		Instruction	Network sniffing & ARP poisoning	
	Transport & App protocols	13/02/2019	2	CLASS	2	L3	Security of Network protocols -	Attacks against TCP. DOS,	
		13/02/2019	2	LAB	2			Free practice with ettercap	
		15/02/2019	2	CLASS	2	L4	Security of Network procols - Application layer	review of attacks at application prot.	
		15/02/2019	2	FP	2				groups formed
	Net discovery	20/02/2019	2	CLASS	3	L5	Network reconnaissance and mapping	Network services, recoinnasance, scanning, and fingerprinting	
		20/02/2019	2	LAB	3		Instruction	Network discovery and scanning	
		22/02/2019	2	PRJFB	3		Project presentation	Presentation of project topics	
		22/02/2019	2	FP	3				
System aspects	Vulns	27/02/2019	2	CLASS	4	L6	Vulnerabilities & attack surfaces	Vuln definitions, code errors, conf errors, vuln repos, vuln examples	Lab project Submit by: 10/4/2019
		27/02/2019	2	LAB	4		Vulnerability assessment	Vuln lab	
		01/03/2019	2	CLASS	4	L7	Penetration testing	Pentesting as a process	
		01/03/2019	2	PRJFB	4		Project feedback	Setup project & feedback	
	Carnival	06/03/2019							
		06/03/2019							
		08/03/2019							
		08/03/2019							
	Social Eng. & pentesting	13/03/2019	2	CLASS	5	L8	Invited lecture: pentesting @ TU/e	exploit testing, automated frameworks	
		13/03/2019	2	LAB	5		Instruction	metasploit	
		15/03/2019	2	CLASS	5	L9	Hacking a human	social engineering	
		15/03/2019	2	PRJFB	5				
	Attacks on the web	20/03/2019	2	CLASS	6	L10	Attacks - web attacks	trust issues, XSS/CSRF, SOP bypass + SSL downgrade + fingerprinting	
		20/03/2019	2	LAB	6		Instruction	Web security CTF	
		22/03/2019	2	CLASS	6	L11	Attacks - malware	review of existing malware	
		22/03/2019	2	PRJFB	6				
	Advanced attacks	27/03/2019	2	CLASS	7	L12	Invited lecture: malware analysis	malware analysis	
		27/03/2019	2	LAB	7		Instruction	Malware analysis tutorial	
		29/03/2019	2	CLASS	7	L13	Attacks - advanced web attacks	malvertising, spam, Drive-bys	
		29/03/2019	2	LAB	7		Demo	Exploit kits	
	Privacy	03/04/2019	2	CLASS	8	L14	Privacy in networks	honest-but-curious, VPNs, TOR	
		03/04/2019	2	PRJFB	8		Project feedback	Setup project & feedback	
	Noclass	05/04/2019							
		05/04/2019							
	Exam. Period	10/04/2019							
		10/04/2019							
		12/04/2019							
		12/04/2019							