
Secure Programming II

(SecProg 2)

Engin Kirda

kirda@eurecom.fr

Welcome to Secure Programming II

- For those who are lost: You currently in the first lecture for the *Secure Programming II* 😊
 - This is an advanced course that aims to make you “security-expert”
 - So far, as a engineers, you have learned to write code and build applications, break them a little... we show you how to break them even more 😊
 - Our aim is to help you learn typical and common security mistakes (i.e., vulnerabilities) by breaking applications
 - Hacking, of course, is illegal ;-)

Why is this course called SecProg II?

- The course has evolved over the years
 - First, seminar on Internet security
 - Then, advanced Internet security
 - Later, at Eurecom, advanced topics on security
 - A little later, secure programming...
 - Now, secure programming 2...
- We cover practical, systems security topics

Administrative Issues

- Mode
 - lectures and regular programming assignments
 - written final
 - we have 14 sessions that have been given to us
 - lectures will take about an hour
 - we will leave you time for assignments
- When and Where
 - Monday, EC 06, 13.30
- Slides and News (please visit regularly)
 - <http://www.iseclab.org/secprog2/>
- Mailing List
 - For questions, you can contact secprog@iseclab.org

Lectures

Tentative Topics

1. Unix Security
2. Windows Security
3. Stack Buffer Overflows
4. Heap Buffer Overflows and Format String Vulnerabilities
5. Intranet Application Security (NFS, NIS, SNMP, CVS)
6. Race Conditions
7. Reverse Engineering and Binary Analysis
8. Malware, Viruses and Worms
9. Botnets
10. Social networkings
11. Advanced web security

Who should do SecProg II?

- People who would like become “security gurus”.
 - We usually took part in a Capture the Flag hacking contest against other universities. Lots of fun: We’ve held 1st, 2nd, 3rd and 4th places in Vienna
 - At Eurecom, we’ll see if there is interest...
- People who are “hard-core” technical
 - C and Linux should not be a problem for you
 - If you have never written a C program... then this might not be for you
- You should be interested in solving technical problems
- You will need to invest “private” time...
 - writing a worm, reverse engineering applications, writing a Trojan

Your Roadmap to Enlightenment

Challenges Solved	Rating
0	Script Kiddie Nobody+ Nobody++ Nobody Junior Nobody Senior Nobody Professional Apprentice Stackmaster InetSec1
1	
2	
3	
4	
5	
6	Apprentice+ Apprentice++ Apprentice Junior Apprentice Senior Apprentice Professional Stackmaster exploit Warlock Guru / Master Guru (CtF required) InetSec2
7	
8	
9	
10	
11	
12	
13	
14	

Lab

- Assignments
 - 8 challenges
 - Starts on the 16th of October, 1 new Challenge every 2 weeks
 - no immediate credit assigned
 - You are required to solve 4 assignments to be able to finish the course
- Environment
 - assignments should be mostly solved at home
 - small test network, which is remotely accessible via ssh (Linux)
 - accounts can be obtained as of 12th of October
 - check home page for details
- Turning in
 - hard deadlines (with sufficient time)
 - automatic checking with immediate feedback

Lab

- Capture the Flag (CtF) Exercise
 - security exercise involving universities around the world
 - Eurecom has participated once... We will see this year ;-)
 - teams have to hack into other machines while simultaneously defending their own systems
 - probably rather involved and time consuming
 - but very rewarding and interesting
 - more information under <http://www.cs.ucsb.edu/~vigna/CTF/>

Lab

- Registration
 - You need to send me an e-mail AS SOON AS POSSIBLE:
 - kirda@eurecom.fr, Subject: SecProg registration
 - I will create an account for you and you will get the information for logging in
- Contacting us (for whatever reason)
 - I don't want to receive e-mails such as "it does not work"
 - We will not debug for you
 - Specify a problem as much as possible

Lab

Assignments (tentative)

1. Unix Vulnerabilities (Input Validation)
2. Windows Security (maybe... ;-))
3. Stack Buffer Overflow
4. Advanced Buffer Overflow (e.g., Heap, Format String Vulnerability)
5. To be determined
6. Forensics
7. Program Analysis and Patching (“Cracking”), maybe malware analysis
8. Simple Malware (writing a virus or a worm)

Time for the infamous DARPA video
and other material ;)

Conclusion

- Remember: send me e-mail for registration
 - kirda@eurecom.fr
- Remember: this is an advanced course
 - If you cannot program, don't do it
- Remember: The labs are time-consuming
 - You are expected to work by yourselves, at home
- Remember: You will learn a lot. However: No pain, no gain ;-)