Software development security 101

Marc Espie <espie@lse.epita.fr>

See https://www.lse.epita.fr/teaching/courses.html

April 7, 2022

There are more conferences for attackers than conferences for safety. That is the problem.

Theo de Raadt

Introduction

Basic goals

- Re-explain the basic ecosystem of software from a security perspective
- Give you enough vocabulary to pass internship questions
- Dispell misconceptions about development security

Advanced goals (for the elective course)

- Modern mitigation and development techniques
- Introduction to source-code review and auditing (from a security perspective)

Prerequisites

Setting limits

- You've mostly done C and Unix so far
- That does always matter
- And a few problems which are not C specific

The fine print

- All your fancy languages have C/C++ runtimes
- Unix has a fine security model

Bibliography

- Building Secure Software (Viega, McGraw, ISBN 0-201-72152-X)
- OpenBSD papers: `http://www.openbsd.org/papers/`
- Ted Unangst's FLAK: `http://www.tedunangst.com/flak/`
- Follow @internetofshit on twitter

How to pass the exam

Multiple choice question

- You will have to know basic terms
- You should be able to RTFM for Unix
- I assume you have the basics concerning C and Unix development, nothing fancy compared to the Piscine or 42sh.
- If you've attended the lectures, you should be able to pass

How to pass the exam 2

Advanced questions

- There will be source code samples
- It won't be 100% clean
- It won't be exactly like "standard epita code"
- If it's different it's not necessarily wrong
- Beware of wrong assumptions
- You should be able to point out the most problematic line

How to pass the exam 3

Feedback from the previous years

- there was a more advanced course
- but over half the students failed the exam...
- ... because of missing prerequisites, like basic C and system
- ... so this year, the advanced course is gatewalled
- ... you *must* pass the systems course and sede 101 to be able to register for the second course
- ... and I will limit the number of students to a manageable number