Secure Programming
Asg1 – Secure Chat

Erik van der Kouwe

October 28, 2024

VU VRIJE UNIVERSITEIT AMSTERDAM

Assignment Goals

- Learn to design a secure distributed application
- Learn how crypto can be used in practice
- Practice applying secure programming guidelines
- Practice low-level programming
- Practice using a real-world cryptographic library

Assignment Overview

- Design and build secure chat client
- Requirements
  - Written in C, runs on UNIX
  - Limited external code allowed (see assignment PDF)
  - Functional requirements (send/recv private/public msg, …)
  - Security requirements (what is attacker prevented from doing)
  - Text-based interface
- Optional bonus addition: web chat

Program Environment

- Program should work on Ubuntu 22.04 LTS Desktop x86_64
- In practice, you can develop in any UNIX environment, assuming that you do not rely on
  - Installed packages not specified in assignment
  - New features not present in reference versions of packages
  - Implementation-defined behavior (should avoid that regardless)
    - Example: integer sizes
  - Undefined behavior (should not do that in any case either)
    - Common source of vulnerabilities, discussed in course

Programming in Windows

- Windows is not UNIX, but can be used for UNIX development
  - Windows Subsystem for Linux (optional part of Windows 10/11)
- Alternative: install Linux inside VM in Windows
  - VMWare Player
  - VirtualBox
- Alternative: use a UNIX system over SSH
  - Real hackers use vi text editor anyways

Groups

- Assignment may be done in groups up to three students
  - Highly recommended due to amount of programming work
- All members responsible for full assignment
  - Set internal deadlines to have time to verify work your group did
- Group grading
  - Independent of group size
  - In exceptional cases, group members may get different grades
  - In particular, students who did not contribute significant part of the code fail the assignment

Source Control

- Use git for source control
  - Include your .git directory when submitting
  - Do not commit large binary files
  - git history may be used to determine whether all group members contributed significantly
- Plenty of free hosting available (github, bitbucket, …)
- Be sure to mark your project private, and only share with group members
- See SPextra1 for git basics if needed

Planning

- Substantial design and programming work
  - Do not wait for classes to discuss everything, start right away (and correct later if needed)
- Start with a design
  - Helps think about security properties before you start coding
  - Avoid writing code that later turns out unneeded
- Define interfaces (network protocol and header files) early to allow independent work on components

Testing script

- We provide `test.py` to test basic functionality
  - Passing all tests is necessary for a sufficient grade
  - Passing all tests is not sufficient for a sufficient grade
  - Focus on getting all tests to work first, and keep testing
- Covers only absolute basics, testing is still your responsibility

Deadlines

- Deadline A: 12 Nov 2024 at 23:59
  - Basic but functional chat framework
  - Feedback on coding style
- Deadline B: 19 Nov 2024 at 23:59
  - Design for use of crypto
  - Feedback on security
- Final deadline: 3 Dec 2024 at 23:59
  - Full program

Grading

- Deadline A – progress – max 1.0 point
- Deadline B – security – max 1.0 point
- Deadline C – full program – max 8.0 points
  - Secure programming guidelines
  - Meeting requirements
  - Code quality
  - Points deducted for errors/warnings/crashes
- Deadline C – optional web chat – max 1.0 point bonus

More Information

- For more information, see assignment PDF on Canvas
  - Specific requirements
  - Includes help on starting with C as a C++ programmer
  - Includes help on required libraries to use
- Read everything before starting

Getting Help (1)

- Please do not hesitate to ask for help whenever you need it
- Canvas discussion forum
  - Questions about lecture material
  - Questions about assignments not revealing (partial) solution
- Mailing list sp@vusec.net
  - All other questions

Getting Help (2)

- Do
  - Tell us what you are trying to achieve
  - Tell us how you tried to achieve it
  - Tell us why you think it did not work
- Do not
  - Share full/partial solutions with other students

Submission

- Assignments must be handed in through Canvas
  - Follow all submission guidelines in assignment, points deducted otherwise
- Deadlines are strict
  - Assignment not handed in in time → failed
  - Cannot make deadline due to personal circumstances outside your control? Request extension **before** deadline

Plagiarism and Generative AI

- Be sure to consult policy on plagiarism in SP00
- Be sure to consult policy on generative AI in SP00