School of Science
and Engineering
University of Dundee

Assignment 1: Applied Cryptography

Deadline for Submission: 9pm on Friday the 18th of October, 2019

Hand in Method: One team member to email a single .zip file named after the team to s.radomirovic@dundee.ac.uk and with **CC to all** team members. Subject line: "[team name] – Assignment 1".

Date Feedback will be Received by: This will be received within the University's 3 week policy which for this assignment is Friday the 8th of November, 2019.

Penalty for Late Submission: One grade point per day late (meaning if a submission is one day late and marked as a C2 it will receive a C3 grade) A day is defined as each 24 hour period following the submission deadline including weekends and holidays. Assignments submitted more than 5 days after the agreed deadline will receive a zero mark (AB).

Percentage of Module: This assignment is worth 10% of this module.

Overview of Assignment

In this assignment you will, as a team (AC31012 students) or as an individual (AC51042 students), be first implementing **in C++** a secure password login (authentication) procedure and then additionally a password login procedure with a covert backdoor. See the file assignment1-details.pdf for details regarding the password file format and hash function to be used.

There are 4 things that your submitted zip file must contain:

1. A file 'login.cpp'. This is the secure password login procedure. Your login.cpp program must
    a. satisfy the functionality requirements R1—R5 (see assignment1-details.pdf),
    b. compile without warnings when the flags -Wall -pedantic -Wextra are used.
    c. hash the submitted passwords with openssl's sha256 hash function.
    d. The source code must be commented.
2. A file 'login-subverted.cpp'. This is the password login procedure with a backdoor. Your login-subverted.cpp must allow you to login as root or any other user on the system without knowing their passwords and also
    a. satisfy the functionality requirements R1—R3 (see assignment1-details.pdf),

   b. compile without warnings when the flags -Wall -pedantic -Wextra are used.

   c. hash the submitted passwords with openssl's sha256 hash function.

   d. The source code must be commented, but the comments may be misleading.

3. A file 'report.pdf'. This PDF file documents the vulnerability in your login-subverted.cpp. It must be no more than 1 page, **list the team name and team members,** and address the following points:

   a. Steps to trigger the vulnerability. How can an attacker log in to a system without knowing the root user's or some other user's password?

   b. What are the bugs/vulnerabilities in the code?

   c. Why do you think are your bugs/vulnerabilities difficult to detect?

   d. Optional, not part of the page limit: A statement concerning the team members' contributions in case that not all team members have made equal contributions.

4. A 'Makefile'. This file compiles both your secure and your subverted login procedures.

This assignment (part 1 of 2) requires you to implement a secure and a subverted login procedure. A part of your assignment 2 mark will be determined by other teams' inability to find a backdoor in your procedure. Your team will therefore also have to consider operational security: Ensure that no one outside of your team has access to your code and your ideas.

In part 2 of this assignment only the teams' subverted login procedures will be distributed **as source code**. You may assume that the other teams will not have access to your secure login procedure nor its source code.

This assignment can be solved in Ubuntu on the QMB Lab PCs. You may use any other computer and operating system, but it is then your responsibility to ensure that your code also compiles on an Ubuntu configuration as found on the QMB Lab PCs.

Marking Scheme

A grade	B grade	C grade	D grade	Fail	Mark
The login.cpp source code is secure, correct, works with hashed password database and is thoroughly commented. *10 to 7 marks*	The login procedure works correctly, i.e., satisfies 1a, 1b, and 1c. *6 marks*	login.cpp compiles but only satisfies two of the three conditions 1a, 1b, 1c. *5 marks*	login.cpp compiles but only satisfies one of the three conditions 1a, 1b, 1c. *4 marks*	login.cpp *not submitted, or it does not compile or does not satisfy any of the conditions 1a, 1b, 1c above.* *3 to 0 marks*	
Comments:					
The login-subverted.cpp source code is works with hashed password database and is thoroughly commented. (Comments are allowed to be misleading!) *10 to 7 marks*	The login-subverted procedure works correctly with the hashed password database and does not produce compiler warnings (i.e. satisfies 2a, 2b, and 2c.) *6 marks*	The login-subverted procedure compiles but only satisfies two of the three conditions 2a, 2b, 2c. *5 marks*	Login-subverted.cpp compiles but only satisfies one of the three conditions 2a, 2b, 2c. *4 marks*	Login-subverted.cpp *not submitted, or it does not compile or does not allow the attacker to authenticate as another user.* *3 to 0 marks*	
Comments:					
The vulnerability is well explained its covertness is well-justified. It is clear that research has been taken into this area and alternatives considered. *10 to 7 marks*	All three sections are adequately described. *6 marks*	1 out of the three mandatory sections (see 3a, 3b, 3c above) is inadequately covered. *5 marks*	2 out of the 3 mandatory sections (see 3a, 3b, 3c above) are inadequately covered. *4 marks*	Inadequate report or no report.pdf submitted. *3 to 0 marks*	
Comments:					
All files (Makefiles, report, cpp sources) are submitted in the required file format and structure. The report is not longer than 1 page. *10 to 7 marks*	Minor issues in one of the submitted files or with the submission (e.g., naming, missing team members in CC, etc.). *6 marks*	One or more files inadequately submitted or minor issues with two or more files. *5 marks*	One file missing. *4 marks*	Two or more files missing. *3 to 0 marks*	
Comments:					
Marks awarded by ranking submissions from shortest to longest in number of ";" and "," used in source code.	Code is obfuscated as, e.g., HEX strings in order to minimize number of ";" and ",".	Sum of ";" and "," characters in source code of subverted login is greater than 65.	Sum of ";" and "," characters in source code of subverted login is greater than 100.	Inadequate subverted login procedure submitted (e.g., implemented w/o hash function.)	

10 to 7 marks	*6 marks*	*5 marks*	*4 marks*	*3 to 0 marks*	
Comments:					