Tartan Secure Camera Application

Overview

Assume that you and your team are working for Tartan Inc. You're readying a new remote camera application designed for both business and personal users. Management wants to release the software as soon as possible, but there are concerns that the new software system is insecure. Before the software is released it must be completed using rigorous secure software development standards and evaluation.

Learning Goals

We will use the Tartan Secure Video Conferencing application as a context for our project. Please note that the purpose of this project is not to make you an expert in network programming, but rather the context is being used as a model problem for you to practice software security techniques and methods presented in the course. Software development is necessary to fully explore these concepts, but programming is not the main focus. There are many dimensions to this project; the major themes include the following:

- Learning how to implement secure software development practices in the context of a team software project.
- Learning how to evaluate the security of an existing software system that you did not develop.
- Developing a plan to manage security on a software project.
- Gaining experience using industry standard software security tools during and after development.
- Learn to use data to validate decisions on a project.

Your team will execute this project in two phases that are described below.

Phase 1. Development

Your team will develop a secure implementation of the system. Specifically, you must design and develop a system to meet the following notional requirements. The application should support two modes of communications: 1) a secure mode where all data is guaranteed to be private and 2) a real-time mode where performance projecting real-time images is the dominating requirement.

The proposed system has the following basic functional requirements. Note

- A user should be able to initiate a video feed, end a feed.

- A user should be able to end a video feed.
- A user should be able to save a video feed for offline review.
- A user should be able to tune image analysis.

The system also has the following architectural concerns (i.e. quality attributes)

- Performance: The system must deliver video as close to real time as possible, especially in real-time mode.

- Authentication:The system must use two factor authentication for sign on and user credentials must be protected. Lost or compromised credentials must be handled in a reasonable way.

- Communication privacy: When in the desired mode the system must ensure that data sent to a user remains private while in transit. No intermediary should be able to snoop or spy on an ongoing video feed.

- Proof of identity (nonrepudiation): Users should be confident that the camera they are using is the one that they believe it is.

- Multi-user privacy: The system must ensure that multiple video feeds remain private between the intended users.

- reliability: The system must ensure that video is reliably delivered. The system should recover from networking errors as soon as possible. The goal is to maintain a secure, performant connection at all costs.

Aside from these requirements, there are a number of basic quality concerns that must be addressed during development.

1. Ensuring that **all software** in both applications are architected and coded to be secure and free of vulnerabilities.

2. Conduct proper fault/error detection, recovery and reporting.

3. Ensure the developed software adheres to the company coding standard and quality standards.

4. Ensure the developed software is adequately tested.

Schedule

The first phase of the project must be completed by the end of the third week of the at-CMU portion of this class.

Deliverables
Phase 1 of the project has the following deliverables:

Preparation. Prior to arrival at CMU you will be assigned a mentor to help guide you through this project. You must form project team; contact mentors; review project description; identify key requirements and standards for new development. Your team must create a project plan to provide rough answers to the following questions:

- How will you organize your team for development? What are your roles and responsibilities?
- What are your milestones?
- What standards and processes will you use during development and why?
- What is your schedule?

This plan should be drafted early on in the project but it is not an official deliverable. Rather you should share it with your mentor and incorporate any feedback.

Phase 1 Deliverables: Source Code, Test Cases, and supporting Artifacts
Your team must assemble a number of artifacts for delivery. Specifically, you must include all source code and for the main system and test cases for evaluation.

Developer Artifacts and Documentation
You must provide a developer guide to enable assessment of your solution. **The developer guide should allow an evaluation team to build and run your system.** Be sure to document major design decisions, especially if they impact security. Be sure to include any and all data collected during development. For example, open defects, external dependencies, and any shortcuts of technical debt incurred. You should also include all non-code artifacts in your delivery, such as requirements and design documentation.

Vulnerability Reporting
You must designate a point of contact to accept vulnerability reports and answer questions about the developed system. You must do this in accordance with best practices. You must make your team reporting policies available so evaluators know how to report problems.

Phase 2: Evaluation
To ensure that no security concerns were overlooked, we will exchange projects to conduct a rigorous security evaluation of the created artifacts. The goal of this evaluation is to identify any and all vulnerabilities in the completed artifacts in such a way that you can make security recommendations to the original development team .

This is a completely different project than in phase one. Accordingly, you will need to reorganize your team to accomplish the task. Your new team organization must be presented to your mentor as soon as phase two of the project begins.

Schedule

The evaluation phase of the project will take place over the final two weeks of the course.

Deliverables

The primary deliverable for this project is a security assessment report. The report must include the following sections:

1. **Executive Summary**: This summary should summarize the major results of your evaluation.
2. **Evaluation Constraints**: What, if any, mitigating factors were considered when deciding how to perform the evaluation.
3. **Evaluation Narrative**: This section should describe in detail the security evaluation activities performed in an organized and understandable way. Specifically, how did you:
    a. Decide which elements to evaluate
    b. Select security evaluation techniques; What techniques were selected and why?
    c. Verify results and prioritize found issues.

    This section should leave the reader with a clear sense of how you approached security evaluation.
4. **Results, Conclusion, and Recommendations:** The evaluation results and all supporting data are described in actionable terms. You should also describe a set of recommendations for the development team to improve security.

Preliminary Design and Artifacts

The secure coding project will involve securing and adding feature enhancements to a set of distributed applications written in the C/C++ programming language running on the Ubuntu Linux operating system. This operating system will be hosted in a virtual machine within Oracle's VM Virtual Box environment. As shown in Figure 1, there will be two applications 1) a Camera and image analysis application and 2) a user display and system control application. These applications shall communicate with each other via an IP network over TCP and UDP.

The camera application will use a USB camera to collect and analyze image frames from the camera using OPenCV's real-time computer vision library (i. e. face or object recognition) as specified. After an image frame has been analyzed the image frame

along with additional amplifying information from the analysis will be transmitted to the user display and system control application.

The user display and system control application is responsible for the following:

1. Establishing communication with the camera and image analysis application as specified.
2. Providing the user interface to control the system as specified.
3. Communicating with the camera and image analysis application as specified.
4. Display image frames and any accompanying amplifying analysis information received from the camera and image analysis application in the format specified.

Figure 1 shows the high-level design of the existing system.

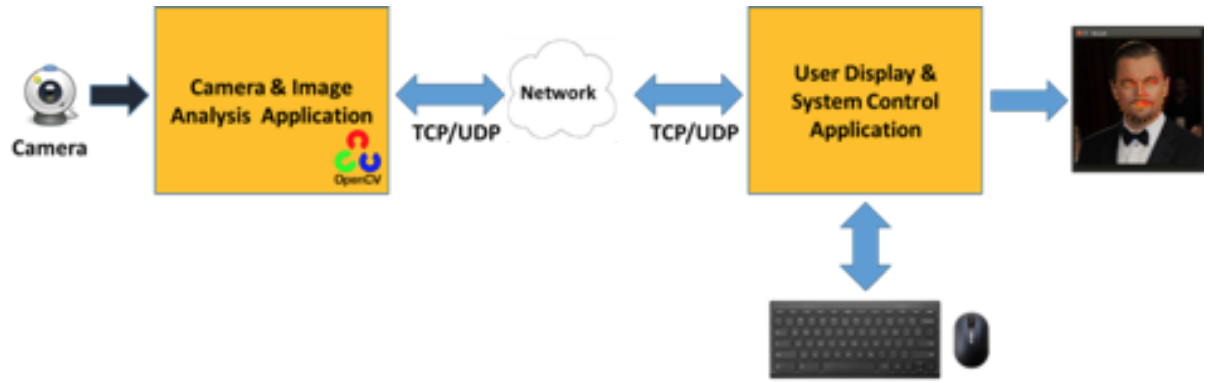

Figure 1: Secure Coding Distributed Application