Term Project

CSE 4481 4.0 Computer Security Lab, Fall 2010

Format: Individual

Project Overview

The goal of the project is to develop a secure chat application that allows secure communication over a public network. The application will be used by non-technical persons for the discussion of classified information. Therefore, the application must be resilient to security attacks as well as user friendly. This chat application will have a client-server architecture whose conceptual diagram is shown on Figure 1.

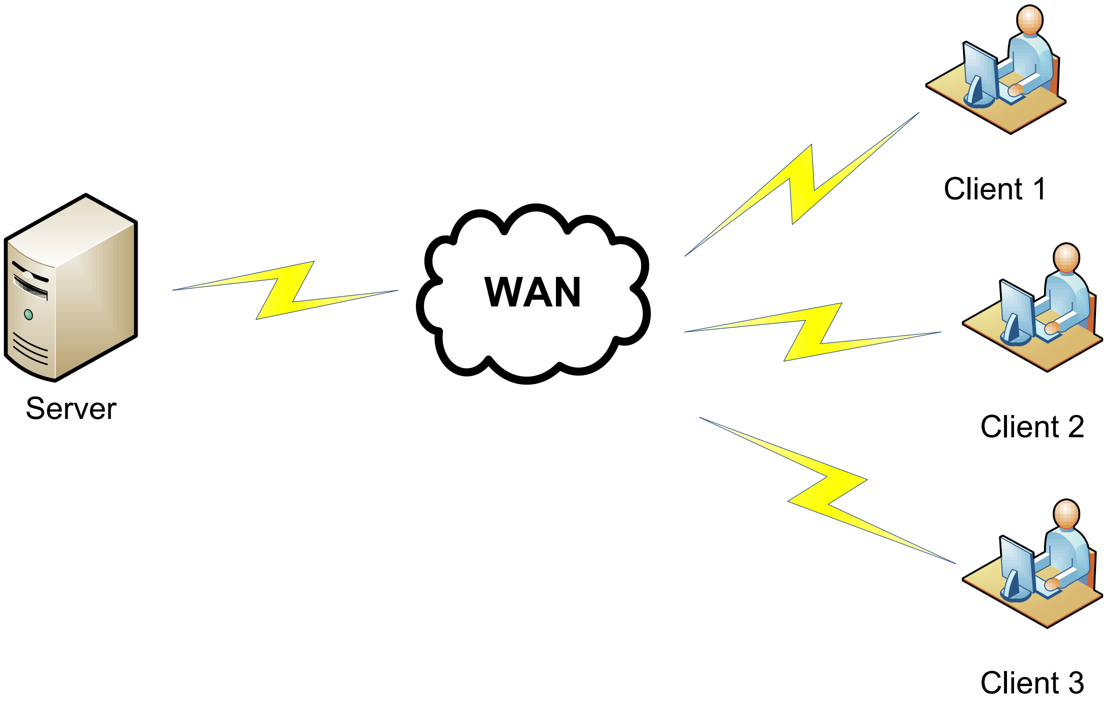

Figure 1: Chat application conceptual diagram

Functional Requirements

The chat application has the functional requirements shown in Table 1.

ID	**Requirements**
REQ_1	**User Authentication**
REQ_1_1	The application shall be able to authenticate users
REQ_2	**Public messages**
REQ_2_1	A user shall be capable to send a public message, i.e. a message sent to all users
REQ_2_2	The sender of a public message shall be displayed
REQ_2_3	The application shall be able to display all public messages
REQ_3	**Private messages**
REQ_3_1	The application shall be capable to send a private message to a selected group of users
REQ_3_2	The application shall be able to display the sender of a private message and the recipients of the private message
REQ_4	**Network Communication**
REQ_4_1	The client communicates with the server using the TCP protocol
REQ_4_2	The server shall bind one TCP port
REQ_4_3	The server TCP port shall be configurable by the user
REQ_5	**Server**
REQ_5_1	The server shall be able to support at least 512 simultaneous users
REQ_6	**Client**
REQ_6_1	The client shall be able to connect to the server using the DNS server name
REQ_7	**GUI**
REQ_7_1	The application GUI shall be **simple** and user friendly
REQ_7_2	The application GUI shall be able to display a list of all logged-on users
REQ_8	**The application shall be secure**

Table 1: Functional Requirements.

Deliverables

The project will be implemented in four phases with the following due dates and deliverables:

Phase 1: Functional application

Due date: October 18, 2010 at 11:59pm

Deliverable: For this phase, you must deliver a fully functional application, i.e. one that fulfills all requirements except REQ_8. You must also provide a report that includes:

- The server and client software design. Your report must include your rationale for the design choices you made.
- The GUI design of your application (includes rationale as above).
- A testing report.

It is expected that your application design will be carefully thought out. An application with inadequate design will be significantly penalized.

Phase 2: Security report

Due date: November 8, 2010 at 11:59pm

Deliverable: For this phase, you must deliver a report that includes:

1. The security test cases (attacks) that were performed (setup, attack code, analysis of collected data, screen shot of collected data)
2. The security test cases that were designed (setup, test case code) but not executed
3. The security design flaws you found
4. The security bugs you found

Phase 3: Secure application

Due date: November 29, 2010 at 11:59pm

Deliverable: For this phase, you must deliver a version of the chat application that contains no known security bugs, i.e. all security issues you identified in Phase 2 have been resolved. You must include a README file explaining how to install your application if you believe this is necessary.

You must also submit a report that includes:

- The revised server and client software design. Your report must include your rationale for the design choices you made.
- The revised GUI design of your application (includes rationale as above).
- A revised testing report. It should include both functional and security tests.
- A report on other ways you could enhance the security of the system, e.g. firewalls, VPN etc.

Phase 4: Final delivery

Due date: December 10, 2010 at 11:59pm

Deliverable: For this phase, you must deliver the final version of the chat application including any fixes based on Phase 3 feedback. You must also submit a revised report based on Phase 3 feedback.

How to Submit

Before the deadline, submit electronically the report and/or the source code you created. Place all the files you want to submit in a directory called PhaseX, where X is appropriately 1, 2, 3, or 4.

To submit, navigate to the parent directory of PhaseX and give the following command

```
submit 4481 PhaseX PhaseX
```

Also, drop off a hard copy of the report (if applicable) into the CSE 4481 assignment dropoff box located on the first floor of CSEB. The hard copy will be the one to be marked. The electronic copy will be used for record keeping.

If the source code you would like to submit resides in the Attack Lab, place it in the following directory in the `Submit` virtual machine: `C:/CSE4481/<cse username>/PhaseX`. You will probably need to create this directory.