Project 2 Description

CS 161 - Joseph/Tygar/Vazirani/Wagner

2nd December 2005

Overview

The goal of this project is to give you some practical experience in analyzing software for vulnerabilities. The project will be due **December 9, 2005**.

This assignment consists of two parts.

Part 1: Protocol Analysis

The goal of Part 1 is to use a protocol analyzer to examine specific protocols in a provided version of the instant messenger program outlined in Project 1.

For this part, you will be using AVISPA (Automated Validation of Internet Security Protocols and Applications). AVISPA is capable of analyzing "large-scale Internet security-sensitive protocols and applications."

1. Visit http://www.avispa-project.org and familiarize yourself with the AVISPA project.
2. Download the tutorial on HLPSL (High Level Protocol Specification Language), the language used by the AVISPA Tool (http://www.avispa-project.org/package/tutorial.pdf).
3. Download the AVISPA user manual (http://www.avispa-project.org/package/user-manual.pdf).
4. You are now ready to use the AVISPA Tool. There is a web interface available (http://www.avispa-project.org/web-interface/) which will allow you to experiment with HLPSL and the AVISPA Tool without having to install anything. Through the web interface, you can select one of the protocols of the AVISPA library, modify it if you like, or write a protocol on your own; you can use one of the four back-ends to check the given protocol, or even use all of them and then compare their outputs. There is also a standalone application available for download on selected platforms. Get a feel for the interface and use the references above to help familiarize you with the AVISPA tool. Look at both the "Basic" and "Expert" interfaces.
5. Read and analyze the TLS (Transport Layer Security) and ISO1 (Public Key Unilateral Authentication Protocol) using the OFMC (On-the-Fly Model-Checker) back-end. These protocols are already encoded in HLPSL and are available via the drop down menu in the AVISPA interface. What does the AVISPA report on these protocols? Explain the attacks, if any, that the tool reports on these protocols.
6. Examine the code and design documentation on the provided instant messenger program. Develop a protocol in HLPSL that reflects the key-establishment/mutual authentication protocol used in the provided instant messenger program. Use the OFMC (On-the-Fly Model-Checker) back-end. What does the AVISPA report on your protocol? Explain the attacks, if any, that the tool reports on these protocols. Propose any fixes in the protocol that are needed.

Part 2: Source Code Audit

The goal of Part 2 is to perform a security audit on the provided instant messenger implementation itself. The following steps are typical of a security audit, and have been adapted from Professor Wagner's graduate course in computer security.

1. Understand the organization, structure, and goals of the provided instant messenger program by inspecting the source code. Draw a diagram depicting the program architecture at a high level. Be sure to include key components and how they interact.
2. Based on your knowledge of your instant messenger program and the diagram from the previous step, state what portions of the program are most likely to have the highest risk of security holes.
3. Analyze the entire source of the implementation, looking for design and implementation errors. This should not be done alone. The purpose of code reviews is to exchange ideas about how the code is written and to establish a standard group interpretation of the code. Simply explaining the code to one another can help increase understanding of the code, and help identify problems.
4. Next, download the Java source code analyzer FindBugs (http://findbugs.sourceforge.net/). FindBugs uses static analysis to inspect Java bytecode for occurrences of bug patterns. The analyzer can be downloaded as a stand-alone application or an Eclipse plugin. Please refer to the online manual (http://findbugs.sourceforge.net/manual/index.html) and FAQ page (http://findbugs.sourceforge.net/FAQ.html) for more information on running FindBugs.
5. Print out the report generated by FindBugs and review the results. Inspect the warnings that the report generates and determine by examination whether the warnings it gives correspond to security holes or not.
6. Write the results of your audit in a formal report, outlining each of the above sections and detailing your analysis of the instant messenger program. For each vulnerability your group determines, you must list the following four items: description of vulnerability (short paragraph), what is required to exploit this vulnerability (1-2 sentences), what would be the impact of an attack against exploit (1-2 sentences), and lastly propose a fix to the vulnerability (short paragraph). Be sure to reference specific methods in your fix proposals.

Project 2 Report

Submit your protocol analysis document and source code audit in a single file using the submit program. Please submit the final document in .pdf, .ps, or .txt formats only. Use the following naming convention for the final documents:

```
CS161-GN_Project2.{pdf, ps, txt}
```

where N is your group number. In the same directory where `CS161-GN_Project2.{pdf, ps, txt}` is located, run:

```
submit Project2
```

Note: You can submit as many times as you want up until the deadline. Only the last submission will be graded. Please submit your project from only one account.