Protocol for Red Team Testing

Matt Bishop, Principal Investigator, University of California, Davis

1. Introduction

The goal of the red team testing is to determine whether there are conditions under which the systems can be compromised. "Compromising" in this context means "tampering or error that could cause incorrect recording, tabulation, tallying or reporting of votes or that could alter critical election data such as election definition or system audit data."[1]

We have two sets of guidelines that are relevant. The first is the 2002 Voting System Standards, which presents requirements for these systems.[2] The second is the Scope of Work (see in particular section 2, "Project Scope and Organization", and Section 5, "Voting System Review Activity and Chronology: Red Team Testing"). Collectively, this plan refers to these as "guidelines". Where they conflict, the Scope of Work governs this test.

Each attack should be tied specifically to a violation of those guidelines. For example, an attack that changes votes stored in a system violates both requirement 2.2.2.1(c) of the 2002 VSS and the prohibition against allowing tampering that could cause incorrect recording and reporting of votes in the Scope of Work.

Because each of California's counties have different procedures to implement the requirements of the law, we may not know the precise environment in which these machines will be used. We therefore may not be able to evaluate the procedures used to protect the machines. To compensate, we will look for attacks that could cause the systems to violate the 2002 VSS. The Secretary of State can then examine state and county procedures to determine if they will deal with those attacks appropriately.

2. Threat Model

The testing assumes two classes of threats: insiders and outsiders. "Insiders" are those who have physical access to *all* components of the voting system, including the election management system. "Outsiders" are those with restricted physical access to the systems, such as voters, poll workers, and observers.

Where system security relies upon proper application of procedures, it may be appropriate to examine the consequences of any failure to follow procedures. For example, if the systems are capable of networking, one might examine what are the consequences if they should be connected to a network in spite of procedural requirements to the contrary; the obvious question is what could happen if someone erroneously connects the system to a network.

Our threat model also assumes that an attacker has access to all details of the system, including source code, and knows exactly how they are used. This provides for those attackers who acquire knowledge over a long period of time, or who have inside access to the system design and implementation (for example, because documents or source code are accidentally posted to the Internet).

[1] Scope of Work, section 5, "Voting System Review Activity and Chronology: Red Team Testing", p. 5

[2] Scope of Work, section 4, "Voting System Review Standards", pp. 3–4.

3. Procedure

Each team leader is responsible for determining how the team proceeds. The following is a suggested approach.

1. Brainstorm attacks. One way is to look at the guidelines and the literature of attacks and claimed attacks, and design tests to determine if the system can be made to violate the guidelines, or if any of the publicly reported attacks work. A second approach is to ignore the standards initially, analyze the documentation and existing literature, and think of ways to compromise an election. Develop attacks from this. Either is fine, as is a combination, or anything else that helps you devise attacks.
   Whenever possible, teams should follow some sort of well-reasoned analysis process to come up with possible flaws. Perhaps the best is to use a methodical tracing of requirements to alleged protection and enforcement mechanisms that then identifies the reference monitor efficacy characteristics of isolation, completeness and correctness. This will help ensure adequate coverage of potential problems. When such a process is used, teams should document what they did.
   The problem with a methodical procedure is, of course, time. Given the limited time period for these systems, we could not do an adequate requirements tracing and test all the potential flaws and attacks. One method would be for the team as a whole to begin this process, and as flaws and attacks are hypothesized, have some members of the team try them (see the following steps). The others would continue the analysis.
2. Prioritize the attacks. The priority in which these hypothesized attacks are to be tried is up to the teams, but in general focus on those attacks that may cause the machine to report inaccurate results or cause the integrity of the ballots recorded to be compromised, and that can be carried out quickly and easily. This is meant as general advice to help prioritize the hypothesized attacks, not to rule out attack scenarios. Each team should use its judgement here.
3. Execute as many of these hypothesized attacks as possible against the systems. Record what happens. As you do this, note whether the attack requires insider access of some kind, and if so what the conditions of access required are. This way, we can classify the attacks as something only an insider can do, or something anyone could do, and identify under what conditions the attack might, or might not, be possible.
4. If possible, identify mitigations. Feel free to note opportunities to improve or harden the system, or recommend changes in future versions of the voting system to enhance the system's ability to meet the guidelines, but also note any *procedural* mitigations that someone could implement to thwart or hinder the attack. This will allow the Secretary of State to suggest procedural mitigations to election officials using the systems.

For each attack, at some point please relate it back to the guidelines. You can do this at any step during the exercise. Some may be easy to relate back when you think of them, but it may also be advantageous to wait until you try it. The attack may not do what you expect!

4. Communication with Other Teams

As team members will sign NDAs for all systems, even those not being examined by their team, the teams can freely discuss any aspect of their work, especially ideas, planned attacks, and anything else in order to come up with new ideas or possible avenues of attack. Teams are encouraged to contact the source code review teams to learn about possible problems they have uncovered, and to pass on possible source code problems.

At least once a week the team leaders will meet together with the red team co-ordinator to update their status, and co-ordinate efforts with each other and the source code reviewers.

5. Documentation

Because we will have approximately 2 weeks of hands-on time for each system, it is unlikely that all attacks can be tried. *Please document all hypothesized attacks, whether they are tried or not.* This will provide a basis for future testing. Please use the following layout for your descriptions, and just fill in what you can as you go along.

Date:
Team:
Attack synopsis:
Which requirement or functionality identified in the guidelines does this violate:
Detailed description of hypothesized attacks:
System being tested; identify the component or components:
Experiment conducted:
What happened:
Interpretation: did the attack succeed:

Please ensure there is enough detail so the attack, *and the results*, can be reproduced.

6. Team Work

The exact method in which the teams work is up to the team leader and the members of the team; the critical requirement is that the report on the system and the vulnerability/attack documentation be completed on time. One suggestion is that the teams spend two weeks working with the systems, and plan to spend the third week documenting what they have found.

7. References

- "Scope of Work", Exhibit A, Agreement No. 06158101 (May 2007).
- *2002 Voting System Standards*, Federal Election Commission (Apr. 2002).