Project: Analyzing a DRE

Introduction

This project focuses on electronic voting systems, in particular those known as DREs (for *Direct Recording Electronic*). These systems are widely used, but often have problems such as miscounting votes or not booting up properly. Some of these problems are human; others are technical; and still others are both. Here, you'll be examining one of the systems used in electronic voting. Your job: figure out what policies and procedures are *necessary* to ensure the machine works properly!

You will be using the Flaw Hypothesis Methodology to do this. This project has several phases.

Phase 1: What Is It Supposed to Do?

An electronic voting system is designed to play a particular role in the election process. *No* part of the election process is perfect. Each part fits into a process intended to minimize errors. So, think of a DRE as replacing the punch cards, paper ballots, level machine, or other mechanism used to record the voter's votes. The DRE can be flawed, but if the other policies and procedures of the election process ameliorate the flaws so they do not affect the election results, the flaws are acceptable. If, however, the flaws cannot be ameliorated, then the question is whether the use of DREs makes the election results at least as reliable as the existing mechanisms.

The first phase of this project is to figure out what an electronic voting system should do. You are to list the requirements that such a system should meet. Examples of such requirements might be: count votes accurately, do not associate names with votes, handle provisional ballots (expand this one by giving more details).

Good places to look for this information are at books about elections, web sites (check out the various Secretaries of States' web pages, or the Clerk-Recorders' web pages for the various California counties), and papers on electronic voting.

What is due. Please turn in a list of the requirements for an election, with justifications for each requirement. List all that you can think of, and for each requirement say whether the requirement affects the DRE. For example, two such requirements might be:

- Ensure only the Egads party members can win: required because by law, no other political party has paid for the election; not a DRE function because the poll workers enforce this by saying there are no key cards for the other parties
- Ensure the DRE breaks down after 4 votes are cast: required because it causes the unemployment rate to go down as people are hired to fix the machines; a DRE function because the DRE must be programmed to fail

When it is due. October 19, 2006

Phase 2: What Problems Might Arise?

Now that you have the requirements, associated with the use of DREs, you need to examine what can go wrong. There are a number of threats that can arise. Your job is to examine the threats and determine which involve the DREs. For example, an election official may misspeak when announcing the results. This is a threat to the accuracy of the election (it's being misreported), but it's not something the DREs can do anything about. As another example, the DRE may miscount the votes. That affects the accuracy of the election, and *is* something the DREs can do better (by counting accurately).Make these threats as specific as possible; this will make the next phase easier. Don't forget physical threats, such as spilling water into the machine!

Again, the sources for phase 1 will help you here. Other good sources include the various studies on electronic voting machines done in the past. Some excellent ones are the Brennen Center study *The Machinery of Democracy: Protecting Elections in an Electronic World* (which contains a well-done discussion of some threats) and the RABA study of the Diebold AccuVote-TS Voting System (which analyzes voting systems in a mock election).[1] Avi Rubin's paper in the *2003 IEEE Symposium on Security and Privacy* is excellent, as is anything by Rebecca Mercuri (of Notable Soft-

[1] Disclaimer: I was involved in these, so I'm probably biased.

ware), Doug Jones (he's a professor at the University of Iowa) and David Dill (a professor at Stanford University). There are also several organizations that have documents that may prove useful. Poke around the web, and you'll find them.

What is due. Please turn in a list of threats, tied to the requirements for an election. Each threat should identify the requirement (or requirements) that the threat will prevent from being satisfied.

As a simple example:

- A non-Egads party member masquerades as an Egads party member, is allowed to vote, and writes in names of unauthorized candidates; this circumvents the requirement that only Egads party members can win
- A DRE does not crash during Election Day; this circumvents the requirement that DREs break down after 4 votes are cast.

When it is due. November 2, 2006

Phase 3: How Can You Test For Those Problems?

Now that you have identified threats, you need to figure out how to test for them. For each threat, come up with an experiment that will determine whether the threat can be carried out, and if so how. The importance of this step is to figure out which threats can be turned into attacks, and how hard it is to either block, or detect and recover from, the attack.

The system you will be examining is the Hart eSlate. We chose this system because it is the one Yolo County will be providing to polling stations to satisfy the requirements in the Help America Vote Act (HAVA). Yolo County plans to provide *one* eSlate per polling station. Most people will vote using paper that will be scanned at the Clerk-Recorder's office. But anyone who wants to use the eSlate can.

The particular system is important because the configuration and layout of the box may suggest attacks.

What is due. Please turn in a list of possible attacks, and how to test to see whether they would succeed. ***You need not actually launch the attack to determine whether it will succeed—in some cases, you can tell whether the attack would work without launching it. For each proposed attack, see if you can come up with a way to test for it without launching it.***

As an example, consider the first threat from the previous example. The DRE component of the threat would require either that ballot images for parties other than the Egads party be stored on the machine, or the Egads ballot images allow write-in votes. For the former, you can check whether there are cards corresponding to other parties; for the latter, bring up the ballot and look for write-ins. This is an example of a testing protocol that does not involve launching the attack.

Now consider the last threat, that the DRE is built to keep running after 4 votes are cast. To test this, you might cast 5 votes, and see if the fifth vote causes the machine to crash. Once it does, you then need to restart the machine, and then try again; perhaps it only crashes after the first 4 votes are cast. This is an example of a testing protocol involving launching the attack.

When it is due. November 17, 2006

Phase 4: What Are the Results of the Testing?

Now, we will have several eSlates available. Your job: figure out which attacks will work, and which ones will not.

For each potential attack, take the testing protocol you devised in the previous phase and implement it. We will discuss in detail the way to do this, but the general rule is to *document everything so thoroughly and completely that anyone can duplicate your test and get the same results*. If it works once but you can't duplicate it, note this. That may mean the failure is intermittent, but unless it can be duplicated, people will question whether it's a problem.[2]

For this phase, assume the attackers have unfettered access to the DRE. This is *very* unrealistic, but we'll get to that in the next phase. For now, your job is to come up with the worst nightmares about DREs that an election official can have, and make them reality.

[2] Your instructor is one such person.

What is due. A list of the tests, your notes, and the results. Remember that saying the attack would fail is a ***perfectly valid*** result! You will not be graded on the number of successful attacks; you will be graded on the quality of your testing. In blunter terms, if Alice does a poor job of testing (cryptic notes, can't duplicate the testing protocol, etc.) but compromises the system, and Bob does a good job of testing (detailed notes, duplicating the testing protocol gives the same results, etc.), Bob's grade will be considerably higher than Alice's.

Remember, the educational goal of this project is to learn how to do penetration testing and to learn the Flaw Hypothesis Methodology. It's not to break a voting system.

When it is due. November 30, 2006

Phase 5: How Can We Fix the Problems We Found?

You now have a set of threats against the DRE that can be realized. But they all assume that the attackers can do whatever they want to the DREs. Here's where reality is to intrude. Few (if any) election officials allow such access to DREs. Your job is to figure out the most effective procedures to stop the attacks you've found. Or, if you can't prevent them, what procedures would enable the election officials to detect the attack, and compensate for it?

For this phase, don't worry about what is *actually* done; instead, focus on what *should be* done. Also, be as complete as possible. For example, if one defensive mechanism requires that a poll worker hear a card being ejected, note that the polling station must be quiet enough to ensure this. (If you want to know where this comes from, read the RABA study mentioned above.)

What is due. For each attack that worked, document what procedures and policies are *required* to prevent the attack from succeeding, or that will allow the attack to be detected and compensated for. In accordance with the principles of secure design, be as restrictive as possible *with respect to the election officials*. For example, requiring all voters to be strip-searched will eliminate some of your attacks, I suspect; but it's completely infeasible because it would turn voters away from the polls.

When it is due. December 8, 2006

Conclusion

The goal of this project is to teach you how to use the Flaw Hypothesis Methodology to analyze a system. We expect you to apply it here, especially in phases 2–4. The context in which you are applying it concerns a system in use, and performing a critical function in society. We hope you learn from this project, and enjoy it.