E-Voting as a Teaching Tool

Matt Bishop

Department of Computer Science
University of California, Davis
bishop@cs.ucdavis.edu

Abstract. Electronic voting systems are widely used in elections. This paper describes using an e-voting system as the basis for a project in an undergraduate computer security class. The goal of the project was to teach the students how to use the Flaw Hypothesis Methodology to perform a penetration study.

Keywords: computer security, information assurance, electronic voting, flaw hypothesis methodology, penetration study

1 Introduction

The mark of a good class is that the topic becomes more than an abstraction. The students are able to translate their knowledge into something tangible, and to apply what they learn in real life. Computer security offers many opportunities to do this. One of the most exciting applications is to electronic voting.

Electronic voting is particularly well suited for an introductory class in computer security. The act of voting is a process that most students are familiar with, and that many have done. But "electronic voting" or, more precisely, the use of electronic voting systems (called "e-voting systems" here), is simply a small part of how elections are run. Elections involve many processes and procedures, including designing ballots, reporting results, managing the precincts and polling stations, and so forth. E-voting systems are designed to replace paper ballots. The theory is that eliminating paper will cut storage costs, make discerning the voter's intent simpler by eliminating ambiguity in marks or "hanging chad", and allow automated counting of votes so that results can be reported more quickly.

The numerous reports of problems with e-voting systems in the media raise questions about the utility, accessibility, and security of these systems. Yolo County uses "optical scanning," in which voters mark paper ballots that are then scanned to count the votes. In order to meet the accessibility requirements of the Help America Vote Act, the Clerk-Recorder used Hart InterCivic DAU eSlate systems, chosen in part because the optical scanning systems were from Hart InterCivic, and in part because they were the most responsive and easiest of the e-voting vendors to work with. The Clerk-Recorder asked if we could have our students examine the systems and report on any specific policies and procedures that should be in place in order to protect the use of the systems—in less precise terms, what did she need to do to keep them secure?

Section 2 describes some necessary background, specifically the flaw hypothesis methodology and some details about how elections are run. Section 3 describes the structure of the project, and section 4 the results of the exercise. The paper concludes with some thoughts on the exercise.

2 Background

We begin with a review of the flaw hypothesis methodology, and then describe how an election in Yolo County works.

2.1 The Flaw Hypothesis Methodology

The flaw hypothesis methodology [1] is a structured technique for performing penetration studies. It is most effective when done in the environment in which the system is to be used; then policies and procedures will affect the results, either for better or for worse.

The methodology consists of four phases.

1. In the *information gathering* phase, the testers analyze the environment and the system to learn as much about both as they can. They learn how the system is deployed, operated, and shut down. They learn about the stated policies and procedures, and how those are actually practiced.
2. Using the knowledge and information obtained from the first phase, the testers then *hypothesize flaws* in the system. They may also draw on their knowledge of related systems, and of the flaws in those systems. Human failings are a valuable source of flaws, because people often do not follow proper procedures—or the procedures for handling unexpected or rare events may not exist.
3. The third phase *tests the hypothesized flaws*. Often, time limits or other constraints prevent all the hypotheses from being tested. Those that are must be documented thoroughly. If the test demonstrates that the flaw does not exist, the testers proceed to the next flaw. But if the flaw does exist, the testers proceed to the next phase.
4. The final phase is *generalization*. In this phase, the testers examine the flaws they have found, and try to generalize them to find other flaws. As an example, if 3 network daemons have similar flaws, a logical generalization is that a library common to all 3 daemons contains a flaw.

We have used this methodology in projects in the past. One class helped test an intrusion tolerant system. The system specification stated that the system's performance would degrade no more than 25% if an attacker gained access to it. Therefore, the goal of the penetration test was to gain unauthorized access to the system and then cause a degradation of system performance by more than 25%. A different class tested a deception mechanism; the goal of the study was to determine the type of system being targeted. In both cases, we used

special control mechanisms to restrict the students so they would not accidentally attack systems other than those involved in the test. For the voting machines, we kept the systems in locked rooms, and did not connect them to any networks. All access required the students to be physically present. This simulated the environment in which the machines would be used, as California forbids them to be connected to a network.

2.2 Elections and All That

Elections have several security-related requirements. Key ones are *accuracy* (the votes are recorded and counted accurately), *anonymity* (no ballot can be associated with a particular voter; this is sometimes called *secrecy of the ballot*), and *secrecy* (no voter can prove how he or she voted to a third party; this prevents selling of votes). California imposes other specific requirements for elections. For example, every e-voting system in California must provide a paper trail that the voter can use to verify his or her vote is recorded correctly. This *Voter-Verified Paper Audit Trail* (VVPAT) is used in recounts and is the official record of votes. In addition, only voting systems that the Secretary of State has certified may be used. The certification process examines both hardware and software. This bans the downloading of last-minute "bug fixes" or enhancements.

Unlike many other e-voting systems, the eSlate is not a self-contained unit. A cable connects each eSlate to a second device, called a *Judge's Booth Controller* or JBC. The eSlates may be daisy-chained, up to 12 per JBC. However, in Yolo County, each JBC has one eSlate because only one eSlate would be present at each polling location. The extra cable, coming out of the eSlate and intended to be connected to another eSlate, is tucked into a compartment at the top of the voting unit under a hood called the *privacy hood*. This hood hides the voter as he or she used the eSlate, so observers cannot tell how the voter votes.

When a voter is to vote using the eSlate, a poll worker uses the JBC to generate a 4-digit access code. The voter enters this code into the eSlate. The eSlate notifies the JBC, and the JBC records the access code as used, so it cannot be reused. The voter is then presented with the appropriate ballot, and votes. At the end of the session, the voter is given a summary showing how the votes were recorded, and the summary is printed on a reel of paper in a printer seated in the unit. If the voter accepts the ballot, he or she presses a button to cast the ballot. The VVPAT is marked accordingly, and the eSlate informs the voter that the ballot has been cast. If the voter rejects the ballot, the paper summary is marked as having been rejected, and the voting process restarts.

The vote is stored both on the eSlate and on the JBC. When recounts are performed, the VVPAT and not the electronic record is used.

3 The Project

ECS 153, "Computer Security", is an undergraduate course that introduces computer security to majors in computer science. Students from other technical dis-

ciplines also often take it. The class covers the basic principles, models, and concepts of computer security and assurance.

A major component of the course is a term-long project, sometimes individual, sometimes a class project, in which students apply many of the principles, methodologies, and technologies discussed in the class. This term, the project was to examine the e-voting system to be used by Yolo County and suggest policies and procedures necessary to ensure that the system works properly. We intended the students to gain a deeper understanding of how to apply the flaw hypothesis methodology, to learn about "black box testing," and to discover how a seemingly simple set of requirements requires a complex balance of technology and procedures to implement a particular task—voting. At no time did we have access to source code, or to the ballot generation or tallying systems.

3.1 Phase 1: Information Gathering

Because security depends upon environment, especially when the environment defines the function of the system (as is true with voting), the students needed to learn some details about how voting works. Further, they had to understand how an e-voting system fit within the context of an election. So, the information gathering phase had to include not just the system but also the environment.

The first step of this phase was to determine what an e-voting system should do in an election held in Yolo County. Their report had to list the requirements for an election and explain whether meeting each requirement would involve the e-voting system. For example, one requirement is that only registered voters vote. The e-voting system, which records and tallies votes, is not involved here because only registered voters can use the e-voting system[1]. A second requirement is that the votes be tallied accurately. The e-voting system is intimately involved with this requirement, because it tallies the votes cast on it.

All students knew an election was supposed to produce a winner or winners, and that how someone voted was to be secret. Few realized the problem of votes being sold, which was one of the major objections to paper trails[2]. So the first step was for the students to do research and brainstorm about what the requirements for an election were.

The second step built on the first. Given the set of requirements, if an e-voting system were to be used, which requirements would be applicable to the system? As an example, the requirement of accuracy is clearly applicable because if the machine misrecords votes, then the results of the election will be inaccurate. But the requirement that the systems be delivered to the polling places on time is not a requirement that affects the e-voting systems; it is instead an organizational requirement under the control of people.

So, the first report for the project had two distinct elements. First, the students had to enumerate a set of requirements for the election, and justify them.

[1] California does not allow provisional ballots to be cast electronically.

[2] All paper trails are protected so the voter cannot take a record of his or her votes away from the machine.

Second, the students had to examine each requirement to determine whether an e-voting system would affect whether that requirement was met.

3.2 Phase 2: Flaw Hypothesis Generation

In this phase, the students thought of possible flaws in the use of the e-voting systems. They used the results of the first phase as the basis for this work. The students examined the requirements relevant to e-voting systems, and developed a set of threats which, if realized, would prevent the requirements from being satisfied. The teams read reports about threat modeling of elections [2, 3]. and earlier studies of electronic voting systems to get ideas [4–9].

During this step, the Yolo County Clerk-Recorder's office gave a demonstration of how the eSlate system worked. This helped the students understand what the system looked like and how it would be used. They also learned how the systems were stored, how they would be distributed, and how the results would be brought back to the county seat and counted.

3.3 Phase 3: Flaw Hypothesis Testing

In the next phase, the students had to develop tests that confirmed or refuted their hypotheses. This required the students to analyze the threats, develop general hypotheses, and then refine them to make them testable.

As an example, one good "high-level" hypothesis was that a voter could vote on the eSlate without authorization from the JBC. But there are many ways to vote on an eSlate without authorization from the JBC. One could look at the access code generated by the poll worker, and quickly enter the booth and use the code before the one to whom the code was given. A bug in the software could allow any random 4-digit number to unlock the eSlate. Someone could guess an access code successfully. In order to test the hypothesis, a more specific hypothesis (or set of hypotheses) must be made.

In some cases, students would not be able to carry out the appropriate tests, particularly when the test required equipment that was not available. For example, one team suggested monitoring the electromagnetic emissions from the voting system to read the votes of the voter in the booth; but we did not have access to the necessary equipment. We encouraged the students to list all the tests they wanted to run, along with how to interpret the results of the tests.

For this exercise, the students were told to assume the attackers had unfettered access to the e-voting system. This meant that attacks such as the Princeton virus [10] or Hursti I and II [9] were fair game. With proper precautions, the likelihood of those attacks being successfully launched can be reduced to any desired probability. So, while unrealistic, this assumption set the stage for the last phase of the project.

3.4 Extra Phase: Remediation

A principle tenet of the way ECS 153 is taught holds that students must learn how to fix problems they find. Hence, they were asked to describe policies and

procedures that would hinder or (ideally) prevent any attacks that exploited problems they found.

4 Results

The students were enthusiastic about the project (in part, one suspects, because of the election in the middle of the term). Some teams focused on physical flaws; others looked for problems relating to the software.

One set of hypotheses focused on disconnecting power or cables to see if the vote totals on the JBC, eSlate, and printed paper could differ. Teams focused on the connection between the printer and the eSlate unit. They tried unplugging the connection at various times in the voting proccess. For example:

Hypothesis: One can vote with the printer disconnected. If so, then one can cast a vote without a corresponding vote being recorded on the paper trail. This creates a discrepancy that will cause a vote not to be counted should a recount occur. The test was to disconnect the printer from the eSlate and then attempt to vote. The result was a failure; the eSlate would not accept an access code unless the printer were connected.

They also examined the connection between the eSlates and the JBC, and other eSlates. One team examined the daisy chaining of eSlates, and discovered that it was possible to reboot the JBC and all eSlates:

Hypothesis: The eSlate can be forced to use battery power throughout the election. Normally, the eSlate draws power from the serial cable connecting it to the JBC. It has an internal battery that is used should the power fail. One team noted that disconnecting the JBC's power, or tripping a switch in a conventional power strip, would have this effect. Another team found that plugging a DSUB-15 NULL terminator onto the end of the extra cable at the end of the daisy chain of eSlates caused the eSlates not to draw power from the JBC, thus running on battery power only. The difference in the attacks is instructive, because the first can be remediated by keeping power strips away from voters. The second is far more difficult to prevent, because a NULL terminator fits into a pocket, and the extra cable is concealed in a drawer under the privacy hood. It would take an observant poll worker to notice the motion of plugging the NULL terminator into the cable. The team suggested that the extra cable be removed to solve this problem. Indeed, the Clerk-Recorder has already requested permission to remove the extra cable, and the vendor agreed.

Some teams examined some software issues, trying to overflow buffers for write-in votes (which failed) or looking at the access codes to determine the difficulty of breaking the pseudo-random number generator:

Hypothesis: access codes can be predicted If access codes can be predicted, then at most one access code can be active at a time. So, for example, if a poll worker issued John an access code a_i, and Jane an access code a_{i+1}, then John can first vote with a_{i+1} (which he knows because he can predict the access code after his), and then vote with a_i. Several teams reported finding regularities

in the sequences of access codes they examined, leading them to conclude the generator was not a cryptographically strong pseudo-random number generator.[3]

All teams concluded that proper policies and procedures would remediate the vulnerabilities they found. For the battery-draining flaw mentioned above, the recommended fix was to disconnect the extra cable. For the access code problem, the recommended fix was to ensure only one access code was active at any time. As stated earlier, in Yolo County, each polling station had only one eSlate. So, the recommended procedure was to issue one access code at a time. When the voter was done voting, the poll workers waited until he or she walked away from the eSlate before issuing the next access code.

Grading for the projects depended on the application of the methodology, and not on the number or type of flaws found. All the teams were very successful in the last two phases. Several teams had problems developing the requirements for an election; to help them, after the first phase, we provided a set of requirements that could be used in the second phase[4]. This "leveled the field", so to speak, so students who had trouble with the requirements could continue onto the next phase. Had this not been done, students on teams that did not develop an appropriate set of requirements would have been unable to develop the threats and meaningful hypotheses. With this common set of requirements, the teams all developed sets of threats that, while different, provided a firm basis for hypothesizing flaws.

5 Conclusion

The goal of this project was twofold. With respect to the class, the students were to learn how to perform a penetration study in a structured, methodical manner. This contrasts to the more popular approach of trying attack tools to see what works. That approach fails in the environment provided for the class, because e-voting systems are specialized systems with requirements not shared by most computers. Hence the students had to develop requirements and test against them in order to be able to determine whether they did, in fact, find flaws.

The second goal was to provide the Yolo County Clerk-Recorder with information about policies and procedures necessary to secure the systems on Election Day. Preliminary results were passed over before Election Day, which was halfway through the term. Interestingly, a number of students signed up to help deliver the eSlates and JBCs to polling places, and act as troubleshooters. To prepare them properly, the vendor arranged a troubleshooting class at UC Davis for the students (and others). The class was a success, everyone feeling that they better understood how the systems functioned and how to fix problems. Of course, this also led several students to hypothesize ways to attack should the recommended procedures not be followed.

[3] None, however, presented the algorithm used to derive the access codes.
[4] We did not say which ones affected the e-voting system, leaving that to the students.

As a result of their involvement in this project, several undergraduate students joined a group of graduate students who are analyzing the e-voting systems. This is an ongoing project, and one we hope will prove useful to Hart InterCivic, the Yolo County government, and ultimately the citizens of Yolo County who cast their vote, expecting it to be recorded and counted accurately.

Acknowledgments

The Yolo County Clerk-Recorder, Freddie Oakley, suggested this project and loaned us the electronic voting systems used in this study. She and Tom Stanionis readily provided information about how they planned to use the eSlates. Greg Hinson gave the class a demonstration of how the eSlates and JBCs were set up and used. Without their help, this project would have been impossible, and we thank them. We also thank Hart InterCivic for holding a troubleshooting class at UC Davis.

References

1. Linde, R.: Operating systems penetration. In: 1978 National Computer Conference, AFIPS Conference Proceedings. Volume 44. (1975) 361–368
2. Saltman, R.G.: Accuracy, integrity, and security in computerized vote-tallying. NBS Special Publication 500-158, Institute for Computer Sciences and Technology, National Bureau of Standards (now NIST), Gaithersburg, MD (August 1988)
3. Brennan Center Task Force on Voting System Security: The machinery of democracy: Protecting elections in an electronic world. Technical report, Brennan Center, 161 Avenue of the Americas, 12th Floor, New York, NY 10013 (August 2006)
4. Kohno, T., Stubblefield, A., Rubin, A.D., Wallach, D.S.: Analysis of an electronic voting system. In: Proceedings of the 2004 IEEE Symposium on Security and Privacy. (May 2004) 27–40 Appeared previously as Johns Hopkins University Information Security Institute Technical Report TR-2003-19, July 23, 2003.
5. Compuware Corporation: Direct recording electronic (DRE) technical security assessment report (November 2003) http://www.sos.state.oh.us/sos/hava/compuware112103.pdf.
6. Science Applications International Corporation: Risk assessment report: Diebold AccuVote-TS voting system and processes (September 2003) http://www.dbm.maryland.gov/SBE.
7. RABA Innovative Solution Cell: Trusted agent report: Diebold AccuVote-TS voting system (January 2004)
8. United States Computer Emergency Readiness Team: Diebold GEMS central tabulator vote database vote modification. Cyber Security Bulletin SB04-252 (September 2004) http://www.us-cert.gov/cas/bulletins/SB04-252.html.
9. Hursti, H.: Diebold TSx evaluation and security alert (May 2006) http://www.blackboxvoting.org/BBVtsxstudy.pdf.
10. Feldman, A., Halderman, J.A., Felten, E.: Security analysis of the Diebold AccuVote-TS voting machine. Technical report, Princeton University (September 2006)