

# Introducing Security Analysis in Computer Security Courses Through an Electronic Voting Project

Chad D. Mano<sup>1</sup> and Aaron Streigel<sup>2</sup>

**Abstract** - For courses in computer security, one of the more difficult skills to impart to the students is the ability to conduct methodical security analyses of computer systems. One must create a system that is sufficiently complex to generate non-trivial security issues while avoiding completely overwhelming the students. To that end, we have developed a new course project based on the design and evaluation of physical electronic voting systems, a problem area that is sufficiently complex, yet relatively easy to understand. In this paper, we will discuss the project with regards to higher level learning objectives (analysis, evaluation) as well as give observations and refinements made through offering the project over multiple years in an upper-division computer security course.

*Index Terms* – computer security, course project, e-voting.

## INTRODUCTION

A typical computer security course is filled with new terms and basic concepts which are necessary to build the foundation for a greater understanding of the role security plays in a broad range computer systems. As such, it is easy for students to fall into the rut of focusing on the required rudimentary knowledge without exercising critical thinking to more deeply understand the problems. As educators it is important to not only pass important knowledge on to our students, but also to help them develop the skills necessary to use that knowledge effectively.

In a computer security course, one of the most important and difficult skills to impart to students is the ability to conduct methodical security analyses of computer systems. A system must be created which is not so simplistic as to devolve into a trivial homework exercise while at the same time the system must not overwhelm students with security issues which may be beyond their understanding. Conceptually, one could view this via Bloom's Taxonomy [1], as attempting to achieve a feasible project at the *synthesis* and *evaluation* levels.

To that end, we have developed a course project based on the design and evaluation of a physical electronic voting (e-voting) system, a problem area which is sufficiently complex, yet relatively easy to understand. The project has been part of the Computer Security course, CSE498U [2], at the University of Notre Dame for multiple years. The e-voting project is

both an interesting topic as well as a timely one, as e-voting systems have recently been utilized in elections throughout the country.

Through multiple offerings, this project has proven to effectively push students to learn at the upper tier of Bloom's Taxonomy by requiring them to both incorporate a broad range of knowledge which they have gained in the course, as well as to discover important system requirements which are not specifically given to them. Furthermore, evaluation occurs through both self-review and peer-review which gives each student an opportunity to learn by evaluating and judging the work of other students in the class.

The remainder of the paper is organized as follows. The following section describes the overall project system and goals. The subsequent section reports the results of the project based on multiple offerings of the course and offers future enhancements. The final section presents a summary of the project.

## METHODOLOGY

In recent years the promise of e-voting systems has gained attention for both its promise and failures [3]-[5]. The basic concept of submitting and recording votes electronically is quite simple, yet the design of such a system can be quite complex. An effective design requires a thorough understanding of a range of computer security concepts as well as the ability to analyze the overall process to ensure that weaknesses do not exist in the system. These characteristics of the e-voting system make this an ideal scenario for a security course design project.

The simplicity of the concept allows students to focus on the design of the system rather than spending time trying to completely understand the requirements. In addition, it allows students to discover additional important features which are not explicitly addressed in the project outline. A project where the overall concept is not as familiar to the students may make it difficult for them to discover "hidden" requirements on their own.

Other systems can be used as a basis for similar course projects, but many do not provide the students with as great an opportunity to develop security analysis skills. Projects such as an implementation of Kerberos authentication or a Public Key Infrastructure (PKI) may provide the students with a deeper understanding of certain security features and characteristics, yet they lack the ability to develop the security

<sup>1</sup> Chad D. Mano, Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 cmano@nd.edu

<sup>2</sup> Aaron Striegel, Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556 astriegel@cse.nd.edu

analysis skills of the students. In addition, projects such as these tend to lead the students towards a deeper understanding of the nuances of specific implementations, rather than develop skills which can be lent to a broad range of problems.

In the e-voting project, students are presented with a simple scenario of a local voting precinct environment. A relatively small and controlled environment such as a single precinct is purposely selected to eliminate the need to address the types of outside attacks which would be prevalent in a large-scale Internet-based voting system. Furthermore, this scenario forces the students to think outside the “electronic box” and consider the physical characteristics of the system, typical of system security audits. This is the type of e-voting system which is being deployed in many places throughout the country and will likely become more common in future years.

As the purpose of the project is to enhance the analytical and evaluation skills of the students, only the following minimal set of required features of the system are given to the students:

- **Data integrity:** The system must ensure that the vote count is accurate and contains only valid votes in the system at all times. In addition, it must be able to be audited to verify correct operation both in test cases and in an actual vote.
- **Anonymity:** The system must ensure that the end vote cannot be traced back to an individual voter or a statistically significant group of voters. Anonymity ensures that a voter can vote without fear of retribution due to how the vote was cast.
- **Authentication:** The system must ensure that only valid voters can cast votes and that only authorized parties can view the results from the voting precinct.

The result of the project is a paper which describes the voting system and shows precisely how the system meets the above criteria. Note that not all of the system needs be electronic, rather both physical and/or human controls may be introduced to the system where a purely electronic system is insufficient. Figure 1 illustrates a sample environment for the e-voting system.

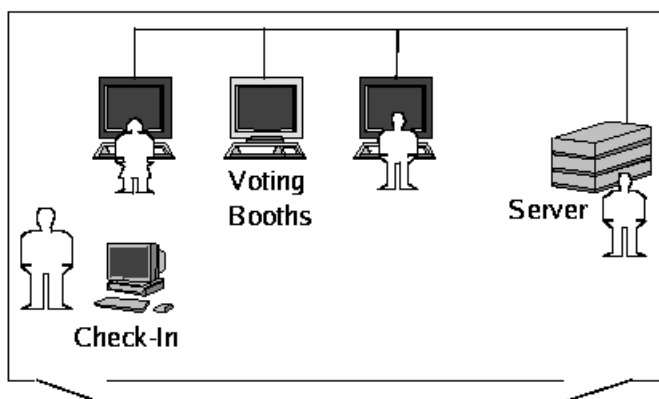


FIGURE 1  
SAMPLE ENVIRONMENT OF AN E-VOTE SYSTEM UTILIZED AT A LOCAL  
PRECINCT ELECTION.

For direction on presenting their solution the students are given the following guidelines for writing the paper.

- *Show how a single vote proceeds through the system.* This leads students to systematically analyze and discuss each security problem and solution in their system.
- *Show how the system can be audited in the event of an error or request for verification of the results.* Similar to the previous item, but applied to the audit capabilities of the system.
- *Show how the system provides authentication.*
- *Show how the system provides integrity for the data.*
- *Show how the system provides anonymity.*
- *For each of the areas, discuss the reasons for selecting the approach including the strengths and weaknesses of the approach.* This requires students to understand the meaning of the three previous items as well as to be able to use appropriate security techniques to ensure them. Discussion of both strength and weaknesses requires students to closely scrutinize their system and understand tradeoffs between possible solutions.

### I. Data Integrity

Data integrity requires the system to ensure the vote count is accurate as well as provide a means for the count to be audited. This requires the students to understand symmetric key encryption mechanisms designed to prevent data tampering during storage, as well as asymmetric key encryption to ensure integrity during data transport. In addition, secure redundant storage or other techniques must be used to provide a fault tolerant method of auditing results.

As part of the system, students should address questions including the following:

- What mechanisms ensure data integrity during data transport?
- What mechanisms ensure data integrity during data storage?
- How do alerts warn officials of a possible data compromise?
- What is the process for auditing data following such an alert?

There are two basic architectures which students will design systems around. Either a stand-alone unit which stores votes locally until the end of the election and then data is sent to a centralized server, or a network-based system where individual voting booths are connected to a server which stores the accumulated data. Each case consists of the stages of storage and transport, albeit in a different order.

For data storage the students should address appropriate encryption techniques which prevent data from being modified in a malicious manner. This requires the students to understand that encryption not only prevents malicious parties from viewing the data, as required by the authentication feature, but also prevents modification of the data, as modified ciphertext, when decrypted, results in invalid values.

As part of data storage, the need for redundant storage of some kind should be addressed. This can be accomplished by

a wide variety of solutions such as multiple storage devices on the voting unit, multiple types of storage devices (flash memory, hard drives, etc.), and storage in multiple locations such as on a local server as well as each individual voting unit. In addition, a physical printout is an important redundancy solution which will be discussed shortly as part of the auditing feature.

Data transport also utilizes encryption key mechanisms to ensure data security. In this case students should address both encryption for the purpose of protecting data values, as well as encryption for the purpose of authentication of senders and receivers. This is different than the authentication feature requirement, in that this refers to authentication between electronic devices such as the voting unit and the collection server rather than the authentication of human parties. This presents the students with an opportunity to incorporate their knowledge of asymmetric key encryption. If the proposed system includes transport of the data from the precinct to a centralized location via modem or some other electronic means, technologies such as Secure Sockets Layer (SSL), IPsec, or Virtual Private Networks (VPN) may be discussed. As part of data transport, the system should ensure that all transfers are idempotent, which ensures that data is not compromised by a system which reports results multiple times due to some human, network, or system based error.

The ability to audit the system is an important requirement for two reasons. First, voters must be assured that their vote is being properly cast. Second, in the event that the electronic vote results are compromised, or there is reason to question the results, a method must be in place to verify the accuracy of the results. In fact, in the 2005 session of the U.S. congress, a bill was introduced which would require e-voting systems to generate a paper-trail for these very purposes [6]. A paper-based audit system is simple to implement, but discovery of this feature requires the students to either, as mentioned previously, think outside the “electronic box” or perform outside research.

In addition to the physical audit system, an electronic system should be employed which can alert election officials of a breakdown in the integrity of the data. This can be done by a method such as comparing checksums of the redundantly stored data, or any other reasonable method which the students may devise.

### *II. Anonymity*

In a traditional paper-based voting environment, anonymity is quite simple to achieve as no direct tie exists between an individual and the paper ballot which is dropped in a box. In an electronic world, it is achievable, but is not as straight forward as a distinct separation between a system which authenticates voters and one which records votes must be made. Students should address questions such as the following:

- How do tradeoffs between providing anonymity and providing auditing capabilities impact the overall system?
- Does the system maintain voter anonymity with the presence of an unscrupulous worker?

Anonymity can be achieved in a variety of ways. The important part is that students thoroughly scrutinize their approach discussing both strengths and weaknesses, and specifically mention how their system prevents a traceback of a vote to a voter. In addition, students should discover and analyze the inherent tradeoffs between the strength of the audit system and the ability to maintain anonymity.

The system should maintain anonymity while allowing election officials to carry out their duties. For example, as part of the auditing system an election official may be required to run reports intermittently. If the official is allowed to run the report as often as he wishes, he may be able to infer the choices of a voters based on the change from report to report. A solution may be to restrict how often the reports can be generated, or have the audit system report only a “pass” or “fail” result rather than actual values.

Providing anonymity may be considered part of authentication in some systems as the physical progression of voting moves from being identified as an authorized voter to the e-voting booth, which should also require some secondary authentication mechanism. Some obfuscation of the link between these two authentication stages can provide voter anonymity. As such, anonymity will be briefly mentioned in the following subsection on authentication.

### *III. Authentication*

This project addresses authentication features in two ways. First, the system must provide a means of ensuring that only valid voters are allowed to cast votes. This can be accomplished in a variety of ways, such as password or even biometric based methods, and allows the students to be creative about the problem and solution. Second, the system must be designed such that only authorized parties are able to view voting results. This requires the students to understand concepts such as separation of privilege, the inference problem, and general database security.

Examples of questions the students should address are:

- Is the voter authentication system sufficient to prevent unauthorized voters from being admitted?
- How does the system separate voter authentication from voting results to provide anonymity while ensuring only authorized voters are allowed to cast votes?
- How does the system prevent unauthorized parties from viewing data while allowing access to authorized officials?
- Can conspiring election officials work together to compromise election results?

As previously stated, voter authentication in an e-voting system occurs in two separate stages as compared to a traditional system which only uses one. In the traditional method, a voter is authenticated by an election official and is then given a ballot. The ballot can be considered a one-time use tool, so no further authentication is needed once the voter enters the voting booth. This is quite different from an e-voting environment.

An e-voting environment requires the equivalent of the traditional authentication method, along with another authentication method at the physical voting booth. The reason is that the voting booth is not tied to the authentication mechanism as is the traditional system which is tied together by the paper ballot issued at the check-in table. In a naïve e-voting system a authorized voter could enter a booth and vote multiple times, which is not possible with a single paper ballot.

First, we address the voter check-in stage. Students could propose a wide variety of methods for identifying authorized registered voters. This can range from an undeviated model of traditional methods to biometric based human identification methods. As this project is part of a computer security course, it should be expected that some form of electronic based verification take place. Creative students may propose systems requiring passwords, smart cards, voice recognition, or other human authentication systems which may have been discussed in the course.

Voter authentication at the physical voting booth should be done in a way that is tied in with the previous check-in authentication while providing anonymity to the voter. In addition, it should prevent a voters from casting multiple votes or perform any other malicious activity. A one-time use smartcard or other identification mechanism given to a voter at check-in which is used to authenticate or activate the voting machine is one solution. This one-time use identification should not be tied to the voter just as a paper ballot does not identify the person who cast it. Again, the important part is that the student make a thorough analysis of the strengths and weaknesses of their approach. As part of the weaknesses, an analysis of potential attacks which a malicious party may attempt may also be important.

On the opposite side of voter authentication there is election official authentication which deals with the protection of data from unauthorized parties. This is part of the physical environment which the students should consider as part of their design. Many election workers with varying levels of authority may have access to individual voting machine units and servers. Depending on the overall system developed by the student, it may be necessary to create a tiered system where workers at different authority levels are able to view varying degrees of sensitive data.

Password-based authorization is an obvious solution to this problem. However, in addition, students may create a custom access control protocol which is specific to the authority structure of their system. An address of the inference problem may be applicable to the security of the system. Many database security mechanisms can be incorporated here to provide the necessary level of security.

IV. Other Issues

There are many other security issues which students may elect to address in their project. Software security topics such as the development process and the procedure for a security analysis of applications could be addressed. Even the method of updating software on a voting machine could be a topic of

discussion as this was an actual problem faced by one company [7].

Other interesting topics include: an analysis of the level of corruption or collusion necessary to defeat the security features of the e-voting system; a discussion of features which may be important to provide public trust of the system; and the implications resulting from the operating system selection for the system. Each of these examples directly relates to the features of data integrity (actual or public perception of), or authentication, but views them from a different perspective than was discussed previously. Students may discover other interesting aspects of the system which may be worthy of discussion.

V. Peer Evaluation

The peer evaluation is a critical aspect of the e-voting project. Evaluation is the highest tier of Bloom's taxonomy and, as such, gives the students a second opportunity to enhance their ability to think critically and methodically through a system security analysis. The project is designed with very general requirements which results in a wide variety of solution implementations. This fact enhances the effectiveness of the peer evaluation because students must analyze a system which is potentially very different from their own design, forcing them to take a new approach in scrutinizing the aspects of this new security infrastructure.

Each student is assigned some number of projects to review and generates a total score which is the sum of the scores for each category shown in Table I. The final grade is based on the peer evaluations as well as evaluations by the TA and instructor with each making up one-third of the final score.

TABLE I  
EVALUATION CRITERIA FOR PEER REVIEWS

Criteria	Description
Presentation	Is the paper easily readable? Did the paper require multiple reads?
Vote Progress	Is it clear how a vote is processed?
Auditability	Is it clear how the system will be audited? How well does the audit system work?
Authentication	Is it clear how the voters/users are authenticated? Will authentication work adequately?
Integrity	Is the integrity of the votes protected? Is the integrity protection sufficient?
Anonymity	Does the system provide anonymity? If not, are the tradeoffs adequate?
Weaknesses	Does the proposal address risks/weaknesses? Does it discuss why/how the design decisions were selected?
Overall	Rating the overall competency of the proposal.

ANALYSIS

The e-voting project has been offered over two iterations of the computer security course (fall 2003 and fall 2004). The makeup for both classes was roughly similar between several graduate students, a few juniors, and the rest consisting of seniors in computer science or computer engineering. Class sizes for the two iterations were 24 in the initial offering and 14 in the second offering. Since the class size was sufficiently

large, students were allowed to split into groups of 1-4 students. Graduate students were expected to complete the project individually.

### I. First Iteration - Fall 2003

During the first iteration (fall 2003), the e-voting problem was given strictly 'as is' without significant in-class discussions. The problem was introduced during part of a single class period and the students were given a full month to prepare a solution. The students were given a handout detailing the requirements and a copy of the review sheet that would be used for evaluation.

From the resulting projects, we made several observations. First, several of the groups narrowed in on obscure problems rather than the overall process itself. For example, one group focused on the physical transport of the ballots themselves after the voting day was done rather than the integrity of the ballots themselves. Other groups made mistakes similar to Diebold [7] by not producing a paper trail nor offering sufficient protection to the data integrity (counters vs. records). Second, the range of sophistication varied significantly between graduate and undergraduate students. While this was not unexpected, it was interesting to note that the juniors in the class were still able to produce quality solutions despite taking a course in operating systems concurrently with the security course.

After the initial results were returned, students were given the option to re-submit an updated version for up to one half of the marked instructor points. Several groups took advantage of this opportunity to repair the above weaknesses identified in the reviews. Third, the final exam illustrated that the students were able to grasp the evaluation concepts. Through both an e-voting evaluation problem (evaluate a system for areas of trust and Denial of Service (DoS) possibilities) and a protocol design problem (integrity of grid computing), the review process was clearly visible in the student thought process (i.e. what questions to ask).

### II. Second Iteration - Fall 2004

During the second iteration of the course, the presentation of the e-voting project was modified to include additional in-class discussions as well as discussions of existing e-vote systems and problems. The emphasis of the project was on the completeness of the process and the audit capabilities of the system. Students were encouraged to think from the perspective of an attacker rather than strictly as a defender. For instance, students were asked, how could they conduct a DoS attack against a given system? Another example would be considering how to bypass the anonymity guards for the system.

Furthermore, the timeliness of the project was especially compelling given the 2004 presidential election. The end result was a significant improvement in the auditability and integrity of the system. However, several students groups still focused on technical issues over the overall process, specifically the technology at the lower layers (software, machines, wiring). Finally, we noticed that students improved

significantly on the final exam with regards to both self-evaluation of protocols and evaluation of areas of trust.

### III. Future Iterations

In order to better improve the project, we are planning on adding the following enhancements:

- *Restrict format*: Rather than defining only a page limit, projects will be required to use a well defined template such as IEEE Transactions.
- *Mandate pictures*: For several groups, the process itself was muddled rather than clearly defined. Although one of the review bullets was to clearly define how a vote proceeds through the system, future projects will be required to draw a picture denoting how a vote proceeds through the system.
- *Restrict hardware*: In order to prevent students from focusing too much on the lower hardware level, the hardware for the system will be explicitly defined for key components (networking, machines, etc.).
- *In-class demo*: An in-class demonstration will be included to show the differences between various auditing mechanisms. Specifically, the demonstrations will focus on the usage of counters versus actual recordings of votes and the implications for DoS attacks.

### CONCLUSION

In summary, we presented our approach for using an electronic voting project to achieve improved student learning in a computer security course. Specifically, the goal of the project was to achieve both *synthesis* and *evaluation* levels of learning, two levels that are critical to students working in security. Through the use of a realistic scenario that is relatively simple to understand, students gain insight into the design of secure systems as well as the evaluation of such systems. Thus, we believe that this project represents an excellent tool for educators in the field of computer security.

### REFERENCES

- [1] Bloom, B. S., Engelhart M. D., Furst, E., Hill, W., & Krathwohl, D., "Taxonomy of educational objectives: The cognitive domain, handbook I", New York: Longmans, 1956.
- [2] "CSE 498U, Computer Security", University of Notre Dame, Department of Computer Science and Engineering, <http://www.cse.nd.edu/courses/cse498u/www/>
- [3] Konrad, R., "E-voting machine crash deepens concerns", *Associated Press*, October 2004.
- [4] Drinkard, J., "Election officials conflicted on electronic voting machines", *USA Today*, May 2004.
- [5] Kohno, T., Stubblefield, A., Rubin, A., & Wallach, D., "Analysis of an electronic voting system", *IEEE Symposium on Security and Privacy*, May 2004, pp. 27-40.
- [6] Holt, R. D., "H.R. 550, Voter confidence and increased accessibility act of 2005", United States House of Representatives, February 2005.
- [7] Jones, D. W., "The case of the Diebold FTP site", <http://www.cs.uiowa.edu/jones/voting/dieboldftp.html>, July 2003.