Trusted Agent Report
Diebold AccuVote-TS Voting System

January 20, 2004

Prepared by: **RABA Innovative Solution Cell (RiSC)**, Dr. Michael A. Wertheimer, Director

RABA Technologies LLC
8830 Stanford Blvd., Suite 205
Columbia, MD 21045

Table of Contents

Table of Contents 2

Executive Summary 3

Introduction 3

Preliminary Findings 5

The Hopkins Report 7

The SAIC Report 11

Red Team Exercise 15

- Smart cards 16
- AccuVote-TS Terminals 17
- GEMS Server 20

Further Recommendations 23

Executive Summary

At the request of the State of Maryland, RABA Technology's Innovative Solution Cell (RiSC) performed a review of the DIEBOLD touch-screen electronic voting system. A team of security experts reviewed the SAIC report commissioned by Maryland and went on to hold a "Red Team" exercise to discover vulnerabilities in the actual voting system as it will be deployed for the March 2004 primary.

The key findings of this effort are two-fold. The State of Maryland election system (comprising technical, operational, and procedural components), as configured at the time of this report, contains considerable security risks that can cause moderate to severe disruption in an election. However, each of these vulnerabilities has a mitigating recommendation that can be implemented in time for the March 2004 primary. *With all these near-term recommendations in place*, we feel, for this primary, that the system will accurately render the election and is worthy of voter trust. However, between the March and November elections we strongly feel that additional actions must be taken to mitigate increasing risks incumbent on a system that will receive broad scrutiny. Ultimately we feel there will be a need for paper receipts, at least in a limited fashion.

Introduction

On November 10, 2003 the Department of Legislative Services, Maryland General Assembly of the State of Maryland (DLS) entered into an agreement with RABA Technologies, LLC to perform a "trusted agent" evaluation of certain aspects of the State Board of Elections plan to use touch-pad "Direct Recording Electronic" (DRE) devices for upcoming elections. The trusted agent role implies that RABA will provide *independent* assessments and will *not seek to profit* from its recommendations. RABA Technologies wishes to thank the State of Maryland for the opportunity to participate in this important project.

The specific requirements of the agreement were:

1. Examine and critique the study conducted by Aviel D. Rubin, known as the Hopkins study.
2. Examine and critique the methodology and practices used by SAIC in its review of the Diebold equipment and the Rubin report.
3. Examine and critique the conclusions reached by SAIC regarding the integrity of the Diebold voting machines and the overall security of Maryland's election procedures.
4. Examine and critique the IT Security Certification and Accreditation Guidelines as issued by the Maryland Department of Budget and Management.
5. Assist DLS in comparing existing SBE practices and procedures to those of the counterparts in other states.

To carry out the work, DLS provided RABA with copies of:

1. *Risk Assessment Report, Diebold AccuVote-TS System and Processes* (unredacted) dated September 2, 2003. This is SAIC-6099-2003-261.

2. *State of Maryland Diebold AccuVote-TS Voting System Security Action Plan* dated September 23, 2003. This document was updated on November 26, 2003 and was downloaded from: http://www.elections.state.md.us/pdf/voting_system_security_action_plan.pdf
3. *State of Maryland Department of Budget and Management, Statewide Security Support IT Security Certification and Accreditation Guidelines*, dated October 25, 2002. This document was prepared by SAIC and may be downloaded from: http://www.dbm.maryland.gov/catalog/opendoc.asp?UserID=2&ID=313005
4. *Analysis of an Electronic Voting System*, by Tadayoshi Kohno, Adam Stubblefield, Aviel D. Rubin, and Dan S. Wallach. This is the so-called Hopkins study (although the last author is from Rice University). To our knowledge this paper has not yet appeared in any journal; it can be accessed from: http://avirubin.com/vote.pdf

During the course of this study numerous other documents proved to be useful. Among these were a report prepared by CompuWare Corporation for the Ohio Secretary of State, the Federal Election Commission's Voting System Standards (approved April 30, 2002), and many news articles and commentaries.

Preliminary Findings

The Hopkins and SAIC reports were given first priority. These will be discussed in more detail below. However, even a cursory examination of these documents (and most of the subsequent public debate) indicated that the analyses were undertaken with less than full knowledge of the *technical*, *operational*, and *procedural* components that must be considered together in assessing any voting *system*.

Furthermore, issues such as voter verifiable paper receipts were largely being discussed either from a security or a cost/benefit point of view, but rarely both. Ground truth on this particular issue is elusive: our analysis can be found below. In summary, we recommend a compromise that should serve to achieve most of the security goals while not being prohibitively costly (in both dollars and people).

Unfortunately, there does not exist at this time a clear set of requirements that any voting system must meet. The Federal Election Commission (FEC), State of Maryland, and NIST[1] guidelines fail to articulate requirements that reflect the unique demands of current all-electronic systems. To quote from the FEC *Frequently Asked Questions About Voting System Standards*[2]:

> **Are the current national voting system standards up-to-date?**
>
> Not entirely. Standards are not permanent. They must evolve alongside technological advancements. Indeed, it is common practice to review and update technical standards every five years or so. The voting system standards, issued in 1990, are no exception to this rule. Vendors are now using new technology and expanding system functions that are not sufficiently covered by the existing standards. Therefore, the FEC is drafting the next version of the standards to cover the newer technology as well as to change standards that currently unduly restrict design.
>
> Still in all, the current standards remain for the most part adequate and useful for the purpose of ensuring the accuracy and reliability of voting systems.

NIST is currently in the process of developing requirements in response to the Help America Vote Act of 2002[3]. In particular it is charged with forming a Technical Guidelines Development Committee that will address, in part, issues of

(A) the security of computers, computer networks, and computer data storage used in voting systems,
(B) methods to detect and prevent fraud;
(C) the protection of voter privacy;
(D) the role of human factors in the design and application of voting systems, including assistive technologies for individuals with disabilities (including blindness) and varying levels of literacy;
(E) remote access voting, including voting through the Internet.

---

[1] National Institute of Science and Technology SP 800-30, Risk Management Guide for Information Technology Systems.

[2] http://www.fec.gov/pages/faqsvss.htm

[3] http://fecweb1.fec.gov/hava/hava.htm

On December 11-12, 2003 NIST sponsored the *1st Symposium on Building Trust and Confidence in Voting Systems*[4]. This conference did not result in any guidelines: it served as a forum for the airing of many constituencies[5] concerns. Until such guidelines are finalized, it will be difficult to perform assessments of voting systems. Maryland is no exception. What is clear is a growing momentum for requiring voter verified paper receipts. To date, California and Nevada now require them. In Congress, Senator Hillary Rodham Clinton has drafted a bill to require a paper trail and security standards for voting machines. Locally, Delegates Karen S. Montgomery (D-Dist. 14) and Joan Cadden (D-Dist. 31) have filed a bill (HB 53) mandating a voter-verified paper trail.

In view of these issues it was determined that a Red Team exercise be held to completely test and stress the exact system to be deployed for the March primaries. This exercise was held on 19 January 2004. For approximately one week prior, RABA's Innovative Solution Cell (RiSC) augmented with consultants from the University of Maryland and U.C. Davis were given copies of the source code and access to both a GEMS server and six AccuVote-TS terminals. The results of this exercise are described in a separate section below. To our knowledge this is the first and only exercise of its kind to determine the security of this system. Much was learned and a list of mitigating recommendations is provided below.

As a final note, RABA Technologies would like to publicly praise the State Board of Elections for their openness, cooperation, and completeness in supporting this study. We were singularly impressed by their willingness to accept "outside" review and to work cooperatively with the single goal of developing the most secure, reliable, and accurate voting system in the country. While there were disagreements about the level of "openness" that should attend this process, this did not hinder our analysis, nor did it bias the results.

[4] For more details on NIST's role in setting voting standards, see http://vote.nist.gov/.

[5] See http://www.nytimes.com/2003/12/15/technology/15neco.html.

The Hopkins Report

A considerable amount of press has been given to the "Hopkins report." The subsequent revelation of a conflict of interest involving one of its authors with a Diebold competitor[6] has only served to detract from the substance of the results. Moreover, many of the statements made by the authors appear to function more as attention gathering "sound bites" than actual statements of fact. Every attempt was made to sort through this posturing and evaluate the specific software vulnerabilities the authors uncovered.

The single most relevant finding in this section is that the general lack of security awareness, as reflected in the Diebold code, is a valid and troubling revelation. In addition, it is not evident that widely accepted standards of software development, such as the Carnegie Mellon Software Engineering Institute's Capability Maturity Model® for Software and System Security Engineering (SW-CMM and SSE-CMM), were followed[7]. Diebold does claim that their software development environment is ISO 9000 compliant.

We generally agree with the conclusions of Hopkins Report on purely technical matters. We further agree with SAIC on a number of mitigating issues that speak to incorrect hypotheses assumed by the Hopkins Report. These are detailed below.

Unfortunately, neither of these reports employs the concept sometimes known as *defense in depth*[8]. Simply put, the security of a system should not be solely assessed as the sum of its components, but rather it should assume that certain precautions and systems will fail. One then measures how well the system contains the damage, alerts to it, and recovers from it. We comment on this in more detail in the SAIC section.

In short, we feel that continual reviews of the Diebold code are both prudent and necessary. Diebold is on public record stating[9]:

> Diebold Election Systems has and will continue to open up its system for review by respectable, unbiased, third-party experts such as those evaluations conducted in Maryland and Ohio. We are confident in the integrity and security of our system, and that the electronic voting format holds the greatest potential for ensuring impartial, secure and accurate elections.

Our attempt to acquire this code, as an agent of the State of Maryland, required a standard non-disclosure agreement. We found Diebold systems to be forthcoming and generally consistent with their public statement.

There are a number of other reviews of the Hopkins report that add depth to the discussion[10]. These are highly recommended for their additional insights and perspectives.

Specific Comments: In the following we quote from and comment on the key Hopkins report findings.

1. **Hopkins:** *"The most fundamental problem with such a voting system is that the entire election hinges on the correctness, robustness, and security of the software*

---

[6] See http://www.jhu.edu/news_info/news/home03/aug03/votehere.html for a statement regarding this issue.

[7] We would expect to see at least level-2 certification of the programmers.

[8] See http://nsa1.www.conxion.com/support/guides/sd-1.pdf

[9] http://www.diebold.com/dieboldes/ohio.htm

[10] See, e.g. Doug Jones' analysis at http://www.cs.uiowa.edu/~jones/voting/dieboldftp.html#rebuttals and Rebecca Mercuri's at http://www.notablesoftware.com/Papers/critique.html

within the voting terminal. […] The only solution to this problem is to introduce a 'voter-verified audit trail'"

Response: A thorough review of the literature on this matter exposes a very rich debate: by no means are these statements held as universal truths [11]. In any system a certain trust model must be posited. The Hopkins report posits threats in the form of unscrupulous voters and malevolent insiders (such as election officials and the developers of the various systems that control the election). We are confused as to how the existence of paper receipts would not be subject to the same threats. For example, unscrupulous voters could easily create their own receipts, substitute them in the ballot box, and then later claim the machines malfunctioned (although there are now suggestions that the receipt should only be viewable behind a window). Similarly, malevolent insiders could theoretically cause the machines to record ballots incorrectly, display and print receipts that the represent the vote cast, and then substitute their own receipts (behind closed doors) to match the fraudulent tally. If the software, processes, and procedures of the all-electronic system are implemented robustly, if the source code and operating systems are subject to rigorous testing, and if the security risk model is continuously and accurately updated it is theoretically possible to drive down the risk to the point that the introduction of voter verifiable receipts is counterproductive.

The debate must center on whether using e-voting systems introduce *additional* vulnerabilities. In particular, the procedural vulnerabilities are mitigated by a separation of privilege: more than one person can monitor the task. For example, in California, parties can send observers to watch the ballot boxes being moved to county seats, or to watch the counting. But that cannot be done with programming the e-voting systems. As an alternative, testing is allowed; for example, in California, the optical counters must be validated by counting the votes of 1% of the precincts manually, the assumption being that this will validate the optical counter. But testing e-voting systems is much more complex, and cannot be validated in the same way. Hence, these systems need to be used in such a way that the controls that apply to other parts of the election process will apply equally well to the e-voting part of the process.

2. **Hopkins:** *"'Security through obscurity' is a long-rejected theory that systems can be made more secure by simply hiding the security mechanisms from public view"*

   Response: Good security principles dictate that the analysis of a system should presume that all components are publicly known. However, that does not imply that it is good practice to make those components known[12]. In a commercial environment one must respect the rights to intellectual property that can provide a competitive edge. It can be argued, however, that subjecting source code to open scrutiny will not only motivate the programmers to write better code, but it will leverage the expertise of a much broader audience. It has the obvious downside of providing the malevolent user a blueprint of the system. Nevertheless, we are not aware of any in-depth source code analysis done on the Diebold software that matches the Hopkins team effort.

[11] A sampling can be found at http://www.aapd.com/dvpmain/votemachines/paperballots.html

[12] For example, the US Government classifies its high-grade encryption systems. This is done not only to make it harder to exploit, but to prevent its adversaries from easily leveraging its best technology. The AES and DES standards, which are in the open source model, are advocated for *less secure* applications.

This is troubling (the SAIC report was undertaken in three weeks and focused on procedural issues; the State of Ohio report was more substantial but was not comprehensive. Others, such as in Nevada, may be in the works). The Independent Testing Authorities validate functionality but do not perform security analyses. In consideration of these facts, we strongly recommend that the SBE require their vendors to provide independent source-code level security assessments for their products. Proprietary concerns should never be allowed to mask security through obscurity.

3. **Hopkins:** *"Even with this restricted review of the source code, we discovered significant and wide-reaching security vulnerabilities in the AccuVote-TS voting terminal. Most notably, voters can easily program their own smart cards to simulate the behavior of valid smart cards…."*
   Response: Since the publication of the Hopkins Report a number of modifications have been made to the AccuVote-TS system. We report on the results of these changes in the Red Team section, below. However, the statement made in the Hopkins study is essentially true *provided fixed passwords are used*. DIEBOLD now provides software to create "Security Key Cards" which election officials can employ to change these passwords, largely eliminating this threat[13]. The RiSC team was able to validate this functionality by generating its own Security Key Card.

Subsequent to the publication of the Hopkins Report, the SAIC review, and Diebold's rebuttal, the Hopkins authors produced a response[14]. This response is useful in addressing what are clearly the most important points. We take these in turn.

1. **Cryptography**: The AccuVote-TS system continues to employ DES for encrypting ballots on the flash memory PCMCIA cards. The default is to use the known key mentioned in the Hopkins Report. As described above, DIEBOLD has provided functionality to change these keys and this is a strong recommendation. For the encrypted link between a precinct and the local board of election, the system now uses AES, dynamic keys, and a standard SSL connection. Keys and passwords are no longer shared between users and procedures have been put into place to manage them effectively. The claims of Rubin et al, while largely mitigated, were very useful in stimulating many improvements in the overall security of the system. These procedures should be codified so that future versions adhere to them. Unfortunately, as the Red Team exercise showed, the issue should have been (and continues to be) *authentication*. Digital certificates only exist at the servers and these are neither signed nor authenticated by the AccuVote terminals. No authentication by the AccuVote-TS terminals is instantiated. As demonstrated in the Red Team exercise, it is readily possible to have a precinct unwittingly download its results (and password) to an anonymous laptop which could modify the results and upload them to the appropriate GEMS server – in real time. Focusing on encryption may have unintentionally diverted attention from the more pressing vulnerabilities demonstrated in the Red Team exercise. More details can be found in that section.

[13] This threat is largely mitigated considering the use of standard smart card readers. The smart cards do not allow more than five guesses at a password. There exists more sophisticated equipment that can probe cards for passwords.

[14] See: http://avirubin.com/vote/response.html

2. **Smart cards:** These are discussed in more detail in the Red Team section. We completely agree that it is prudent to assume that the Diebold source code is available to an adversary. The scenario that permits a voter to insert a wiretap device between the voting terminal and the smart card is theoretically valid, but certainly unlikely. The smart card reader is internal to the AccuVote-TS device so this complicates, but does not nullify, the attack. In either case, it is fair to assume that a smart card can be replicated *provided the passwords used to secure them are known*. We agree that cryptographic protocols would add a layer of valuable mitigation to these attacks and would strongly recommend they be implemented.
3. **Multiple votes:** If an adversary succeeded in producing smart cards that allowed multiple votes, this would be detected by the final reckoning done between the number of voters who cast votes at the specific terminal and number of votes tallied. The fact that voters deposit their voter identification cards at the terminal (into a locked box) allows individuals to be recalled for a revote. It is neither impossible, nor unprecedented to recall entire precincts for a revote. Moreover, a voter verifiable paper receipt does not necessarily mitigate this risk. Indeed, just as an individual might cast multiple electronic votes, it is conceivable they could generate multiple receipts and deposit them illegally. Which receipts would then be used for a recount? The risk of multiple voting requires (as does the current paper vote system) additional safeguards. There are, however, scenarios in which vote buying can be made possible with the ability to manufacture voter cards.
   For example, assuming the ability to program a voter smart card, one could proceed as follows
   a. The vote-buyer provides someone with a voter smart card programmed to simulate a "used" card[15].
   b. The person, whose vote is being bought, then enters his precinct and identifies himself correctly to a voting official. He thereby receives a valid voter card.
   c. That person then pretends to vote (does *not* insert his card into the AccuVote-TS terminal).
   d. Upon leaving, that person turns in the "used" card to the election official (the card given to him by our vote-buyer and created to simulate one which had been used to vote).
   e. The valid card is pocketed and given to the vote-buyer.
   f. The vote-buyer confirms it is a valid card *set to pre-voting status*. He then enters the precinct and votes both the card he legitimately receives from the election official and the one he "purchased" in this scheme.
   g. The counts are exactly correct at the end of the day (two votes cast, two people registered). However, the vote-buyer managed to vote twice – in effect recreating the paper ballot chaining tradition of times past.

[15] Subsequent to casting a ballot, the AccuVote-TS terminal changes the contents of the voter card so that it cannot be used to vote again unless reprogrammed by an election official. Each precinct is provided a number of programming devices for this purpose.

4. **Software Engineering:** The claim that C++ is neither memory safe nor type safe is misleading. The use of the C++ can certainly introduce catastrophic memory vulnerabilities and there are now languages that handle primitives more securely (as the authors point out). However, it is not accurate to portray these other languages as safe. Good programming principles, irrespective of the programming language, are critical. We agree, in principle, that this can be achieved and enforced through independent testing.

Summary: The Hopkins Report must be applauded for its thorough, independent review of the AccuVote-TS source code. It raised numerous valid issues that have resulted in considerable improvements in the overall system. Had the authors approached the State Board of Elections with their preliminary findings, many of their false hypotheses could have been corrected and the discussion not diluted by specious claims. Just as the authors correctly point out that there are standards by which the development of a secure system may benefit, they chose to disregard the similar standards by which systems must be evaluated.

It is imperative, within the confines of proprietary disclosure, that continual, independent testing be done on all software-based election systems. This testing must be comprehensive. It must also be driven by standards, requirements and methodologies vetted in the public domain.

The SAIC Report

From August 5, 2003 through August 26, 2003 Science Applications International Corporation (SAIC) performed a risk assessment of the Diebold AccuVote-TS voting system. Their report is divided into three sections:

1. **Management Controls**: fundamental principles that are inherent in the protection of information systems to manage risk.
2. **Operational Controls**: protection mechanisms that are primarily planned, implemented, and monitored by people.
3. **Technical Controls**: software and hardware based systems that provide security, accuracy, and accountability.

Each section enumerates a number of Baseline Security Requirements. Each requirement is then assessed for whether it has been met, partially met, or unmet, its likelihood/impact, risk rating, and mitigation strategy. The methodologies used are clearly spelled out and conform to NIST and State of Maryland guidelines[16].

The basic findings of the report detail a number of high-risk vulnerabilities across all three categories (management, operational, and technical). The report also presents a number of specific responses to the Hopkins Report.

Interviews with officials at the State Board of Elections indicate that the Baseline Security Requirements do not tie directly to any generally agreed-on guidelines. Indeed, as mentioned earlier, the lack of rigorous guidelines by either the FEC or the National Association of State

[16] National Institute of Science and Technology SP 800-30, *Risk Management Guide for Information Technology Systems*; State of Maryland, *Certification and Accreditation Guidelines*.

Election Directors (NASED)[17] is very troublesome. Our best understanding of the guidelines employed by SAIC is that they reflect internally generated requirements (as developed over similar assessments) together with State of Maryland general IT security and accreditation guidelines.

We were disappointed to see that the FEC Voting System Standards[18], approved April 30, 2002, did not appear as explicit requirements in the SAIC report. Maryland is a signatory to these guidelines and a number of its target goals have not been properly assessed. The FEC guidelines have been well crafted and provide the best requirements baseline we know of for DRE systems. For example, the guidelines require

1. Security
    a. Provide security access controls that limit or detect access to critical system components to guard against loss of system integrity, availability, confidentiality, and accountability.
    b. Provide system functions that are executable only in the intended manner and order, and only under the intended conditions.
    c. Use the system's control logic to prevent a system function from executing if any preconditions to the function have not been met.
    d. Provide safeguards to protect against tampering during system repair, or interventions in system operation, in response to system failure.
    e. Provide security provisions that are compatible with the procedures and administrative tasks involved in equipment preparation, testing, and operation.
    f. If access to a system function is to be restricted or controlled, the system shall incorporate a means of implementing this capability.
    g. Provide documentation of mandatory administrative procedures for effective system security.
2. Accuracy
    a. Record the election contests, candidates, and issues exactly as defined by election officials;
    b. Record the appropriate options for casting and recording votes;
    c. Record each vote precisely as indicated by the voter and be able to produce an accurate report of all votes cast[19];
    d. Include control logic and data processing methods incorporating parity and checksums (or equivalent error detection and correction methods) to demonstrate that the system has been designed for accuracy; and
    e. Provide software that monitors the overall quality of data read-write and transfer quality status, checking the number and types of errors that occur in any of the relevant operations on data and how they were corrected.
3. Non-catastrophic Error Recovery

[17] http://www.nased.org/

[18] http://www.fec.gov/pages/vssfinal/vss.html

[19] This is central to the entire debate.

a. Restoration of the device to the operating condition existing immediately prior to the error or failure, without loss or corruption of voting data previously stored in the device;
b. Resumption of normal operation following the correction of a failure in a memory component, or in a data processing component, including the central processing unit; and
c. Recovery from any other external condition that causes equipment to become inoperable, provided that catastrophic electrical or mechanical damage due to external phenomena has not occurred.

4. Integrity
   a. Protect, by a means compatible with these Standards, against a single point of failure that would prevent further voting at the polling place;
   b. Protect against the interruption of electronic power;
   c. Protect against generated or induced electromagnetic radiation;
   d. Protect against ambient temperature and humidity fluctuations;
   e. Protect against the failure of any data input or storage device;
   f. Protect against any attempt at improper data entry or retrieval;
   g. Record and report the date and time of normal and abnormal events;
   h. Maintain a permanent record of all original audit data that cannot be modified or overridden but may be augmented by designated authorized officials in order to adjust for errors or omissions (e.g. during the canvassing process.)
   i. Detect and record every event, including the occurrence of an error condition that the system cannot overcome, and time-dependent or programmed events that occur without the intervention of the voter or a polling place operator; and
   j. Include built-in measurement, self-test, and diagnostic software and hardware for detecting and reporting the system's status and degree of operability.

Several other categories, such as Audit, Accessibility, Vote Tabulating, etc. are similarly expressed. Our purpose in enumerating these is two-fold: first, there are many critical requirements beyond those that capture public attention, and second, the SAIC report does not provide sufficient scope to assure these requirements have been met or addressed. (We have asked the State Board of Elections to provide a written list of responses to these requirements).

The following assessments are made solely within the scope of the SAIC report. We find the 169 management baseline security requirements to be competently evaluated. Of the thirty-five requirements that SAIC labeled either unmet or partially met, the SBE has taken action to rectify them all. However, documentation of these actions has not been collected into an easily reviewable form. To wit, requirement M-67 states:

> *The system security planning should include information about the type of security evaluation performed, who performed the review, the purpose of the review, the findings, and the actions taken as a result.*

This requirement is labeled 'met', although SAIC notes that their risk assessment is the first performed on the AccuVote-TS voting system. Consequently, the last clause could not be

assessed at that time. Our assessment is that it has only been partially met. We wish to add, however, that SBE officials have demonstrated good faith in meeting this requirement, but the documentation is scattered.

There are 110 operational baseline security requirements. Of these fifteen are labeled unmet or partially met. Several are denoted high risk. We agree with the SAIC report that the key operational needs are security awareness training for election site officials, well-documented procedures for maintaining the integrity of all hardware and software systems, and the ability to detect, and recover from, security breaches in a timely manner. Not only must these high-risk vulnerabilities be mitigated, it is imperative that they be independently reassessed before the system is put into use. They are critical to the legitimacy of the electronic tally.

There are forty-seven technical baseline security requirements. Of these fifteen are labeled unmet or partially met. Since SAIC was unable to perform a thorough source code review, several of the requirements they deem to be met rely on the presumed integrity of the DIEBOLD software and the Microsoft operating systems (Windows CE on the touch-screen terminals, and Windows 2000 on the server). For example, requirement T-8 states:

SBE will ensure security controls are implemented to prevent introduction of data for a vote not cast by a voter.

The report claims this requirement is met because: *"Only voters with a valid Voter Access Card can cast votes."* The presumption is that a valid card can only be issued by an election official. There are numerous scenarios in which valid cards might be forged from knowledge of the source code alone. (This is addressed in the Red Team section).

In one case a requirement (T-34) is declared not applicable (the system should prompt users to change passwords before they expire) because password changing is not enforced. However, T-28 stipulates that the operating systems should enforce password changing. Since T-34 is deemed not applicable, there is no procedural mechanism to assure that it is met once T-28 is addressed. While this example has minor security implications, it does serve to expose the coordination essential in performing a technical evaluation[20].

The technical evaluation spills over into Appendix B with responses to a selection of Hopkins Report claims. It is here that we feel SAIC's technical evaluation is manifestly subpar. For example, SAIC claims in several places that multiple vote casting by an individual is largely mitigated by the openness of the voting booth and by the loud sound the smart card makes when ejected from the AccuVote-TS machine. We see absolutely no validity in this claim: the sound is neither loud nor distinctive nor does it provide any useful security. We further object to discounting buffer overflow attacks by virtue of servers not being connected to a wired network. This certainly mitigates certain access scenarios, but it does not fundamentally address the vulnerability.

In general, the SAIC technical responses rely on the integrity of the code as installed on the AccuVote-TS terminals and the security procedures put in place. No accounting is made for the failure of any of these systems. The level of cascading damage to the system is never assessed *presuming component failure*. Good security practice (defense-in-depth) acknowledges that systems do fail and constructs layered defenses to prevent or at least

[20] Indeed, SBE officials were forced to go through an additional round of testing when initial DIEBOLD code rewrites (to address the SAIC unmet requirements) caused critical functionality loss. Once the functionality was restored, security testing had to be rerun.

mitigate the subsequent damage. The SAIC report provides no mitigation strategies for component failure, especially at the software level. As a result, the AccuVote-TS system may be vulnerable to as-yet undetermined threats. Fortunately, the precinct voting systems serve only temporary duty: they exist for a brief (approximately fourteen hour) period in which concerted attacks must be planned and executed. Nevertheless, approximately *sixteen thousand* terminals will be deployed and configured across the state. To assume fidelity is maintained throughout the system is unwise and probably unlikely.

The updated SBE Voting System Security Plan indicates that the changes necessary to meet the SAIC recommendations are completed. While they have procedures in place for acceptance testing of any code changes, it is not evident that any independent oversight is in place to guarantee the accuracy of their claims. Complete, centrally located documentation should be available for independent audits.

Red Team Exercise

On January 19, 2004 a Red Team exercise was held at RABA headquarters in Columbia, Maryland. The members of the team and brief biographies follow:

- **Dr. William Arbaugh**: Assistant Professor of Computer Science at U. Maryland, College Park. Ten years experience at NSA and coauthor of *Real 802.11 Security: Wi-Fi Protected Access and 802.11i* published by Addison Wesley. Research interests are primarily in Information Systems Security and privacy.
- **Dr. Matt Bishop**: Associate Professor of Computer Science at U. C. Davis. Author of a widely-used textbook, *Computer Security: Art and Science* (Addison-Wesley-Longman, 2002). Research interests are in vulnerability analysis and code analysis. Consulted for the Yolo County, California, Clerk-Recorder on electronic voting issues.
- **Mr. Paul Chakravarti**: Director of RABA Strategic Solutions Group. 15 years experience working for the Government Communications Headquarters (U.K. counterpart to the NSA). BS in Applied Physics with particular expertise in software development, integration, and security.
- **Mr. Paul Franceus**: B.S. in Computer Engineering. Ten years experience at NSA with networking, distributed systems and computer security. Last eight years at RABA.
- **Mr. Mark Mclarnon**: B. S. Computer Science. Two years at Booz Allen, Three years NIST, one year in public sector performing information security. Employed at RABA since October 2003.
- **Mr. John Ormonde**: BS in Electrical Engineering and MS in computer science. Fifteen years at NSA working Computer Security, 1.5 years at Booz-Allen, and three years at RABA.
- **Mr. Shawn Smith**: B.S. in Electrical Engineering (U. MD.) Had several software engineering jobs before joining RABA in 1998. Has worked in the security field for five years developing software to test the security of computer systems.

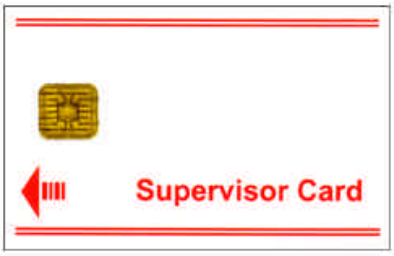
- **Dr. Michael A. Wertheimer**: Director of RABA Innovative Solutions Cell (**RiSC**). Twenty-one years experience as cryptologic mathematician at the National Security Agency, last three years as senior technical director for the agency. Joined RABA in August 2003.

A Red Team exercise is designed to simulate the environment of an actual event, using the same equipment and procedures of the system to be evaluated. Teams are then free to experiment with attack scenarios without penalty. In this fashion a broad range of vulnerabilities may be discovered and validated in an operational environment.

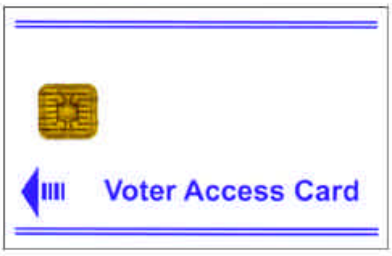
The team focused on smart card vulnerabilities, AccuVote-TS terminal security, GEMS server security, and the methods used to upload results following an election. Since the scope of the effort was contractually limited (in dollars) the team's focus was necessarily directed toward the most likely vulnerabilities. Indeed, the total software package contains roughly 285,000 lines of source code, only a fraction of which could be carefully studied. In all cases the team first approached the system with no foreknowledge of the source code. As the attacks became more sophisticated, an increasingly in-depth understanding of the actual system was necessary. In the following descriptions every attempt is made to point out precisely what must be known in order to exploit the vulnerability being described.

Smart cards

Initially, the team was only aware of two types of smart cards: Supervisor cards and Voter Access Cards. These cards contain microprocessors capable of performing cryptographic and password functions[21]. Both card types are identical in model number; they differ only by the actual information stored on them and their exterior labeling. The information on these cards is password protected; in other words, it is not possible with a *generic card reader*[22] to read them without first providing the correct password. While it is likely that more sophisticated probing equipment can discover this information, Red Team members were able to guess these passwords. Indeed, the passwords used to protect both types of smart cards provided to the team appear in the source code that the Hopkins team evaluated. *Initial guesses on the team's part provided instant access to the cards' contents.*



Given access to the cards' contents it became an easy matter to duplicate them, to change a voter card to a supervisor card (and vice versa) and to reinitialize a voter card so that it could be used to vote multiple times. While time did not permit a demonstration, the team did map out the procedures and hardware necessary to load this capability onto a pocket-sized PDA. Starting from scratch, we estimate the cost to be under $750. The team did,



[21] See http://www.cardlogix.com/product_smart_select_most.asp for a product description and pricing.

[22] Card readers/programmers are readily available through Internet sales. The model used by RiSC was obtained for $29.95 + overnight shipping.

however, demonstrate this functionality with a generic laptop.

The use of hardcoded passwords is surprising both as an inferior design principle and in light of them being published openly in the Hopkins report. It must be assumed these passwords are well known.

Included with the system are Voter Card Encoders. Election officials use these devices to enable Voter Access Cards. Specifically, an Encoder writes information to a Voter Access Card that will cause the AccuVote-TS terminal to bring up the correct ballot for the particular voter. It also sets the password of the Voter Access Card (currently to the default mentioned above).

Further analysis of the source code indicated functionality for a third type of card, a so-called *Security Key Card*. This card allows election officials to change the default passwords used to secure the supervisor and voter cards. The process of changing these passwords is painless and takes approximately five seconds per AccuVote-TS terminal and Voter Card Encoder. The Red Team converted a Voter Access Card into a Security Key Card and demonstrated its functionality. With such a card it is possible to greatly reduce the possibility of multiple voting, card duplication, supervisor card creation, etc. Moreover, the team demonstrated that Security Key Cards could be created (using a laptop) so that no human being would ever know the passwords. Subsequent to the Red Team effort, DIEBOLD representatives pointed out to the team that Security Key Card functionality - and software to implement it – was provided to the SBE several months prior, for just this purpose. SBE has confirmed this fact.

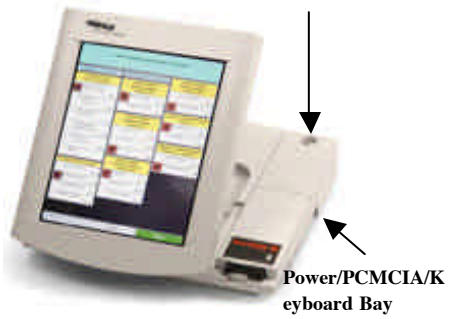
The contents of these cards are neither encrypted nor digitally signed. Thus, for example, the PIN associated with a Supervisor Card[23] can be read directly from the card – provided the password is known. This means creating Supervisor Cards is a simple task: a perpetrator could program his card with an arbitrary PIN that the AccuVote-TS would readily accept.

Immediate Recommendations:

1. Create Security Key Cards with computer-generated passwords *by precinct*. Update all the Encoders and AccuVote-TS terminals within each precinct.
2. Apply Tamper Tape (more on this in the next section) to AccuVote-TS terminals to prevent non-authorized entry of Security Key Cards into the terminals.
3. Institute strict procedures to prevent the use of unauthorized Supervisor Cards.

AccuVote-TS Terminals

The AccuVote-TS terminals consist of both hardware and software components that interact to register ballots. There are clear vulnerabilities in each that the Red Team discovered. A quick description of the terminal is in order.



[23] When a Supervisor Card is inserted into an AccuVote-TS terminal a 4-digit PIN is requested. Given the PIN, one has the ability, e.g., to end the election, clear the vote counts, or vote multiple times.

The hardware consists of a touch-screen voting terminal with two locked bays. One bay houses a roll of paper tape that prints out the initial ("zero count") vote tally and, following the election, the final vote tally. A second bay houses the on/off power switch, two PCMCIA slots (one for the flashdisk card that holds the ballot definition and records the voters' cast ballots and one for a modem), and a standard keyboard jack.

During an election these bays are locked. Maryland has ordered approximately 16,000 AccuVote-TS terminals each equipped with two locking bays and supplied with two keys accounting for 32,000 locks and keys. Surprisingly, *each lock is identical and can be opened by any one of the 32,000 keys*[24]. Furthermore, team members were able to have duplicates made at local hardware stores. It is a reasonable scenario to assume that a working key is available to an attacker.

To make matters worse, using a commonly available lock pick set, one team member picked the lock in approximately 10 seconds. Individuals with no experience were able to pick the lock in approximately 1 minute. Arguably it would be very noticeable for a novice to pick the lock protecting the PCMCIA cards and keyboard interface during an election. However, someone with slightly more experience could do so from a standing position, or two individuals working together could possibly cause enough of a distraction that the lock could be picked while a judge's back is turned. Access to the PCMCIA bay during an election could render the votes cast there unusable. Since certain precincts are considering loading their terminals the day before the election, one must consider this vulnerability as being accessible prior to the election. Among the attacks discovered and demonstrated were:

1. Attach a keyboard to the terminal and access functionality in the software that allows the attacker to view the entire directory tree on the machine's internal memory and on the PCMCIA card. Three functions are then available to the attacker: "Save As", "Finish Recording", and "Open". These functions are remnants of test code that allowed the developers/testers to record events that simulated voting. The <Save As> function along with the <Finish recording> functions allow an attacker to overwrite files in the internal memory and the PCMCIA card. Using this method an attacker can overwrite both the results file and the audit file on both the internal memory and the PCMCIA card. This would completely overwrite the results for that voting terminal. Moreover, the exploit elevates the attacker to Supervisor status - no smart card required.

2. Remove the PCMCIA card. This grants an attacker access to all of the files on the PCMCIA card. Since the only files that have cryptographic protection are the results file and the audit file , the team was able to demonstrate the ability to switch two candidates and still successfully load the election and ballot. Consequently, the voter appeared to vote for the candidate of his choice but he actually voted for another candidate. When writing a shorter candidate name over a longer name it just needs to be padded with spaces to the same length. To write a longer candidate name over a shorter candidate name an rtf format tag was deleted to make the overall length the same. In this fashion a voter can be deceived into thinking he is voting for one candidate when, in fact, the software is recording the vote for another candidate.

[24] This was true of the six units supplied for the Red Team exercise. The State Board of Elections confirmed this fact.

3. Load a PCMCIA card with an update file. The PCMCIA card can be used to update the software on the AccuVote-TS terminal. This can be done by placing a PCMCIA card with an update file into the terminal and rebooting the terminal. The update file allows an attacker to overwrite any file on the system. Furthermore, by using this technique an attacker can install his own version of the ballot station software giving him the ability to completely invalidate all the results on that terminal. If he compromises the AccuVote-TS terminal used as the accumulator[25], he can compromise the entire precinct results.
4. Install new passwords. When the AccuVote-TS terminal is powered on without the PCMCIA card in its slot, the user is prompted to either insert the PCMCIA card or to insert a Security Key Card. As described earlier, it is possible for an attacker to create a Security Key Card due to the use of fixed passwords. An attacker could create a Security Key Card (with a password known only to him), insert it into the AccuVote-TS terminal, and change the passwords. That terminal would then reject all Voter and Supervisor Cards until it could be reset with the correct passwords.
5. Jam the card reader. The Red Team invited people without prior knowledge of the system to vote with an eye towards mischief. One individual was able to crash an AccuVote-TS terminal by repeatedly inserting a disabled Voter Access Card into the card reader. Eventually, the terminal failed to respond and had to be rebooted. This did not cause any data to be lost, but it required the bay to be unlocked and the terminal powered down and back up again.
6. Disconnect the monitor. Other mischief-makers demonstrated the ability to physically tilt the monitor of the AccuVote-TS forward to expose the connecting wires. They were then able to disconnect these wires without causing any damage (essentially just pulling on them). To reconnect the wires into their harness requires opening up the terminal – a procedure that is not allowed during an election.

These are a sampling of the vulnerabilities found as a result of poor physical security coupled with software that fails to use robust encryption and authentication. The team feels confident that physical access to the bay housing the PCMCIA cards, power switch, and keyboard jack can ultimately lead to devastating results to the particular terminal. Most of the near-term recommendations focus on securing this bay.

Immediate Recommendations:

1. Secure physical access to the voting terminal. The team recommends the use of serial-number tamper tape placed both inside and outside each locked bay[26]. This could be accomplished in the following manner: after the terminals are loaded and a zero-tape printed the

[25] Each precinct will designate one AccuVote-TS terminal as the one to transmit the results to the LBE. Following the election, this terminal will be put into "accumulator mode." This allows it to load the results of the other terminals in the precinct.

[26] Two layers of tape are recommended to thwart someone who simply wants to disrupt the election by tampering with the outside layer. If they fail to gain access, the internal tape will verify this and the election may proceed.

bay doors would be secured with the tamper tape and the serial numbers would be recorded. The tamper tape would need to be inspected periodically as a matter of procedure. Ultimately it would be recommended to place alarms on the bay doors.

2. Remove the test recording software from the AccuVote-TS terminal that allows the keyboard exploit. It serves no valid function.
3. Investigate the legal implications of tampering with the hardware systems (such as jamming the card reader and disconnecting the monitor). We see no short term fix for these attacks aside from the clear posting of rules that indicate consequences of such actions.

GEMS Server

Each Local Board of Elections (LBE) will have a Dell computer server running the GEMS software to accumulate and tally precinct inputs. The State Board of Elections (SBE) will also have a GEMS server to accumulate the LBE inputs. To mitigate remote access attacks against these computers, procedures require they not be connected to any outside network. However, to accommodate preliminary (unofficial) results they are equipped to receive counts from precincts (in the case of LBEs) and counties (in the case of the SBE) via telephone modem transfer. The team focused on accessing these computers through their modems. They also considered physical access – but under the guidelines that such access would be fleeting (perhaps 5 minutes to a half hour).

The team also verified that the current version of the GEMS software still contains many of the vulnerabilities widely published on the Internet. It was disappointing to see that no obvious attention was paid to addressing these weaknesses.

The team demonstrated the following vulnerabilities:

1. The GEMS server lacks several critical security updates from Microsoft. As a result, the team successfully exploited a well-known vulnerability using a software product known as Canvas[27]. This vulnerability, described in a security advisory from Microsoft[28] for which a patch was made available on July 16, 2003, allows a remote attacker to get complete control of the machine. Since this is the same weakness that the August 11, 2003 "Blaster" worm exploited, it means that if the GEMS server was exposed to an environment where "Blaster" was propagating, it might have been infected. By successfully directing Canvas at the GEMS modem interface, the team was able to *remotely* upload, download and execute files with full system administrator privileges. All that was required was a valid phone number for the GEMS server.
2. Modify GEMS software and/or election database on LBE server. Given physical access to the server, one can insert a CD that will automatically upload malicious software, modify or delete elections, or reorder ballot definitions. The problem is that the server enables the "autorun" feature.

[27] http://www.immunitysec.com/CANVAS/

[28] MS03-026, http://www.microsoft.com/security/security_bulletins/ms03-026.asp

USB Flash Drive

3. Exploit the USB port in the rear of the device. The back panel of the GEMS server is not protected. Given physical access to a running device it is possible to insert a USB flash drive and upload malicious software onto the server. USB flash drives are easily hidden and can store upwards of 1 GB of data.
4. Boot off a CD. By removing the front panel of the server (this is held in place by a small keyed lock), one can insert a CD, power up the server, and have it boot its operating system off the CD. This gives the attacker complete control over the device. Moreover, the database files that contain the election definition (and results) are neither encrypted nor authentication protected. Results can be modified at will. In addition, ballot definitions can be altered so that the mapping between candidates and their "ordinal numbers"[29] can be changed. A sophisticated user can automate this procedure requiring only a few minutes access to the server.
5. Modify election database. Given either physical or remote access (see below) it is possible to modify the GEMS database. Because both the database password and audit logs are stored within the database itself it is possible to modify the contents without detection. Furthermore, system auditing is not configured to detect access to the database.
6. Social Engineering/Phone line hijacking: The procedure by which precincts upload votes to their LBE is vulnerable to a "man-in-the-middle" attack. This is a result of an incomplete implementation of the SSL[30] protocol. Specifically, the team demonstrated how a laptop could act as a GEMS server. If one could convince the precinct judge to dial into an attacker's laptop[31] then that laptop would not only receive the election results, it would be able to acquire the name and password to access the GEMS server. With this name and password in hand, the attacker could upload modified results to the GEMS server – all in real time. A more subtle attack might involve modifying the settings in the AccuVote-TS terminal (using any of the methods described above) to redirect outbound phone calls to the attacker's computer, or actually gaining access to the phone switch at either the precinct or the LBE.
7. Patches and Updates. The team identified fifteen additional Microsoft patches that have not been installed on the servers. In addition, the servers lack additional measures (all considered best practice) for defense such as the use of firewall anti-virus programs as well as the application of least privilege, i.e. turning off the services that are unused or not needed. Each of these represents a potential attack vector for the determined adversary. While many of these attacks are mitigated by the

[29] Ballot names are presented to the voters, but their votes are recorded as a vote for candidate numbered one, two, three, etc. This exploit can make a vote for the "first" candidate be recorded, e.g., as a vote for the "third" candidate.

[30] Secure Sockets Layer. This is an encrypted protocol that securely transmits data across a network. Unfortunately, this implementation fails to enforce authentication. Hence anyone can impersonate a GEMS server if they can convince the AccuVote-TS to dial into their computer.

[31] For example, one might call the precinct judge and convince him that a modem problem at the LBE requires he change the phone number in the AccuVote-TS terminal. A more sophisticated attack would be to gain access to the phone switch at the voting place or at the LBE and tap into the line. This is actually quite easy to do with equipment that can be purchased for around $1500.

isolation of these computers from networks, one must be mindful of the activation of modem interfaces during the election as well as physical access to the devices before and after the election. Lacking proper software authentication, the State will have to rely on well-trained election officials and judges to strictly adhere to the recommendations given below.

The quantity and quality of the attacks described above is disturbing, especially given the short time the team had access to the system. Certain shortcuts were taken (e.g., assuming knowledge of the phone number that connects to the GEMS server) but we feel these are reasonable actions that a determined attacker would be able to overcome.

Immediate Recommendations:

1. Install all known security patches from Microsoft on the GEMS servers.
2. Ensure modem access to GEMS is enabled *only when uploads are expected*, i.e. via voice notification over a telephone line between a precinct judge and a designated LBE official. The number used for this purpose should be *guaranteed* not to change. Validate the number being dialed and the identities of the callers. The line should remain open until both sides indicate the upload is complete. Shut off the modems when not in use.
3. Turn off all services and ports except those explicitly required by the GEMS software. For defense-in-depth install firewall software to block all ports except those required by the GEMS software.
4. Update the anti-virus software.
5. Turn off services that are not needed by GEMS.
6. Install Tripwire[32] on the system to provide an audit capability on the configuration.
7. Disable the "autorun" feature in Windows 2000 (change the value from 1 to 0 for the registry key: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\CDRom).
8. Ensure the front panel on the server is locked and the server is stored in a physically secure location. Apply tamper tape to the input devices and the reboot button.
9. Change the boot order to make the hard drive first AND password protect the BIOS to prevent changes to the boot order without physically opening the server.

[32] See http://www.tripwire.com/products/servers/index.cfm for one commercial application.

Further Recommendations

The following recommendations reflect observations the team made to improve the GEMS and AccuVote-TS software systems. Given the amount of work this implies, it would be impossible to achieve success in time for the March primary and we question whether it is possible to achieve in time for the November elections.

It is our opinion that the current DIEBOLD software reflects a layered approach to security: as objections are raised additional layers are added. True security can only come via established security models, trust models, and software engineering processes that follow these models; we feel that a pervasive code rewrite would be necessary to instantiate the level of best practice security necessary to eliminate the risks we have outlined in the previous sections. Our analysis lacked the time and resources to determine if DIEBOLD has the expertise to accomplish this task.

Many have advocated the use of paper receipts as a way to provide an audit trail and to validate the accuracy of the electronic systems. This debate cannot be held without considering the basic need for voter trust in the system. While it is our belief that a secure system without paper receipts can be built, it would require not only better software, but also a higher level of sophistication and understanding by those who run our elections. It may never be possible to administer 16,000 autonomous touch screen terminals, 44+ servers, 32,000 locks and keys, thousands of voter cards, supervisor cards, and security key cards with perfect fidelity. As this report indicates, there are many issues to address.

On the other hand, one of the stated Federal Election Commission goals is to have less than 1 in 2 million votes counted incorrectly with electronic systems. Such accuracy has never been obtained with paper ballots in any instantiation. Ballots can be misread, smudged, lost, stolen, destroyed, etc. Furthermore, voters cannot know how their paper ballots will be read; a properly configured electronic system provides this assurance. Thus, the introduction of paper receipts will almost certainly cause a discrepancy between the tallies. Whatever method is chosen to arbitrate this discrepancy, it is certain that accuracy between votes cast and votes counted will suffer.

In discussions amongst the team members, there was no single consensus recommendation, except that the introduction of voter-verifiable paper receipts is absolutely necessary in some limited form. The number of software vulnerabilities such receipts mitigate, the amount of savings they introduce by lowering the procedural requirements, and the trust they garner are likely to be just as cost effective in the long run as a fully locked-down all-electronic system.

However, we do not see the need to install such receipts on every device. Indeed, if all AccuVote-TS terminals are checked to ensure they are functioning correctly before an election, and if they are loaded with identical, digitally-signed, software which is checked both before and after an election, one can make the case that reconciling the results of a single, randomly selected, terminal with its paper receipts is sufficient to believe that the overall electronic counts in that precinct are accurate. Thus, if all the terminals are software and hardware enabled for receipts, one need only provide receipts for a small number of randomly chosen machines. Voters might even be given the choice of using such a terminal – or not. *We make this recommendation only in the face of implementing the software changes given below.*

In any case, we cannot recommend strongly enough continual software, hardware, and procedural reviews – by independent experts – of all voting system components. We further implore NIST, under its HAVA charter, to rapidly develop voting system standards and security accreditation guidelines. Voting systems are a critical infrastructure component of democracy. They should be expected to meet strict guidelines.

General Recommendations:

1. Utilize a smart card as the authentication token rather than a user name and password for uploading results to the GEM server and for accessing the GEMS software on the server. Communications (under SSL) between an AccuVote-TS terminal and a GEMS server must have two-sided authentication (with unique certificates) to prevent man-in-the-middle attacks. The dial-up (PPP) authentication currently uses PAP (password authentication protocol). We recommend CHAP (challenge-response authentication protocol).
2. Develop a base line configuration and audit tools to ensure compliance with the base line.
3. Eliminate group accounts and establish individual user accounts on the GEMS server.
4. Apply and enforce Access Control Lists (ACL) to the GEMS software and databases.
5. Enable system auditing to record and save ACL events.
6. Digital signatures of operating system and GEMS files should be calculated and securely stored off-line (Tripwire was mentioned above).
7. GEMS and AccuVote-TS software contain code for "weighted ballots." This does not have any apparent use for Maryland elections; vendor should provide plausible scenario for weighted-ballots in a general election; if none can be provided, this code should be removed.
8. Establish procedures for off-line updating of all security related patches and virus definitions.
9. Employ a database system with more advanced features than Microsoft Access so that the password and audit log are stored separately from the database.
10. Do not allow software updates without authentication. At a minimum, demand smart card authentication to update the software on the AccuVote-TS terminal. Ideally the software should be digitally signed and verified before installation.
11. Place file protections on all of the files on the PCMCIA card. At a minimum they should be encrypted and digitally signed with keys that are strong and not known.
12. Software integrity should be verified before and after the election to make sure no tampering has taken place. (This can be thought of as “tamper proof tape” for the software.) Currently there is no ability to validate AccuVote-TS software after it has been loaded onto the terminal. Post election validation must be enabled.
13. SBE and LBE system administrators should consider formal security training such as that provided by SANS[33]. Server security templates should then be applied to GEMS servers.

[33] See http://www.sans.org/ : “SANS provides intensive, immersion training designed to help you and your staff master the practical steps necessary for defending systems and networks against the most dangerous threats - the

Ways to implement these recommendations have been discussed by the team. It is our opinion that they can largely be accomplished with currently available technology.

ones being actively exploited." Courses can be tailored to Windows 2000. This is but one of several organizations that provide this service.