DRAFT:
GRAPHIC LAYOUT OF
PRINTED VERSION
MAY DIFFER

THE MACHINERY OF DEMOCRACY: PROTECTING ELECTIONS IN AN ELECTRONIC WORLD

BRENNAN CENTER TASK FORCE
ON VOTING SYSTEM SECURITY,
LAWRENCE NORDEN, CHAIR

VOTING RIGHTS
& ELECTIONS SERIES

BRENNAN CENTER
FOR JUSTICE
AT NYU SCHOOL OF LAW

THE MACHINERY OF DEMOCRACY: PROTECTING ELECTIONS IN AN ELECTRONIC WORLD

THE BRENNAN CENTER TASK FORCE
ON VOTING SYSTEM SECURITY

LAWRENCE NORDEN, CHAIR

VOTING RIGHTS
& ELECTIONS SERIES

BRENNAN CENTER
FOR JUSTICE
AT NYU SCHOOL OF LAW

www.brennancenter.org

ABOUT THE TASK FORCE

In 2005, the Brennan Center convened a Task Force of internationally renowned government, academic, and private-sector scientists, voting machine experts and security professionals to conduct the nation's first systematic analysis of security vulnerabilities in the three most commonly purchased electronic voting systems. The Task Force spent more than a year conducting its analysis and drafting this report. During this time, the methodology, analysis, and text were extensively peer reviewed by the National Institute of Standards and Technology ("NIST"). The members of the Task Force are:

Chair
Lawrence D. Norden, Brennan Center for Justice

Principal Investigator
Eric L. Lazarus, DecisionSmith.

Government Experts
Dr. David Jefferson, Lawrence Livermore National Laboratory and Chair of the California Secretary of State's Voting Systems Technology Assessment and Advisory Board
John Kelsey, PhD, NIST
Rene Peralta, PhD, NIST
Professor Ronald Rivest (MIT), Technical Guidelines Committee, Election Assistance Commission
Howard A. Schmidt, Former White House Cyber Security Advisor for George W. Bush; Former Chief Security Officer, Microsoft

Academic Experts
Professor Matt Bishop, University of California at Davis
Professor David Dill, Stanford University
Professor Douglas W. Jones, University of Iowa
Joshua Tauber, PhD, formerly of the Computer Science and Artificial Intelligence Laboratory at MIT
Professor David Wagner, University of California at Berkeley
Professor Dan Wallach, Rice University

Private Sector Experts (Commercial and Non-Profit)
Georgette Asherman, independent statistical consultant, founder of Direct Effects
Lillie Coney, Electronic Privacy Information Center
Jeremy Epstein, Cyber Defense Agency LLC
Harri Hursti, independent consultant, former CEO of F-Secure PLC
Dr. Bruce Schneier, Counterpane Internet Security
Matthew Zimmerman, Electronic Frontier Foundation

ABOUT THE EDITOR AND TASK FORCE CHAIR

Lawrence Norden is an Associate Counsel with the Brennan Center, working in the areas of voting technology, voting rights, and government accountability. For the past year, Mr. Norden has led the Brennan Center's voting technology assessment project. He is the lead author of *The Machinery of Democracy: Voting System Security, Accessibility, Usability, Cost* (Brennan Center forthcoming 2006) and a contributor to Routledge's forthcoming *Encyclopedia of American Civil Liberties.* Mr. Norden edits and writes for the Brennan Center's blog on New York State, www.ReformNY.blogspot.com. He is a graduate of the University of Chicago and the NYU School of Law. Mr. Norden serves as an adjunct faculty member in the Lawyering Program at the Benjamin N. Cardozo School of Law. He may be reached at lawrence.norden@nyu.edu.

ABOUT THE BRENNAN CENTER

The Brennan Center for Justice at NYU School of Law unites thinkers and advocates in pursuit of a vision of inclusive and effective democracy. The organization's mission is to develop and implement an innovative, nonpartisan agenda of scholarship, public education, and legal action that promotes equality and human dignity, while safeguarding fundamental freedoms. The Center works in the areas of Democracy, Poverty, Criminal Justice, and Liberty and National Security. Michael Waldman is the Center's Executive Director.

ABOUT THE VOTING RIGHTS & ELECTIONS SERIES

The Brennan Center's Voting Rights & Elections Project promotes policies that protect rights to equal electoral access and political participation. The Project seeks to make it as simple and burden-free as possible for every eligible American to exercise the right to vote and to ensure that the vote of every qualified voter is recorded and counted accurately. In keeping with the Center's mission, the Project offers public education resources for advocates, state and federal public officials, scholars, and journalists who are concerned about fair and open elections. For more information, please see www.brennancenter.org or call 212-998-6730.

This paper is the second in a series, which also includes:

Making the List: Database Matching and Verification Processes for Voter Registration by Justin Levitt, Wendy Weiser and Ana Munoz.

Other resources on voting rights and elections, available on the Brennan Center's website, include:

> *Response to the Report of the 2005 Commission on Federal Election Reform* (2005) (co-authored with Professor Spencer Overton)
>
> *Recommendations for Improving Reliability of Direct Recording Electronic Voting Systems* (2004) (co-authored with Leadership Conference on Civil Rights)

ACKNOWLEDGMENTS

Most importantly, the Brennan Center thanks NIST and its many scientists for devoting so many hours to its extensive and thorough peer review of the analysis and report. The report, in its current form, would not exist without NIST's many important comments and contributions.

In particular, we thank John Kelsey of NIST for the substantial material and ideas he provided, which have been incorporated into the report and the report's attack catalogs. We also specially thank Rene Peralta for his original contributions and analysis. Finally, we are enormously grateful to Barbara Guttman, John Wack and other scientists at NIST, who provided material for the attack catalogs, helped to develop the structure of the report, and edited many drafts.

We are also extremely appreciative of Principal Investigator Eric Lazarus's enormous efforts on behalf of this report. His vision, tenacity, and infectious enthusiasm carried the team through a lengthy process of analysis and drafting.

A special debt of gratitude is also owed to election officials throughout the country, who spent many hours responding to surveys and interview questions related to this report. In addition to team members Professor Ronald Rivest and Dr. David Jefferson, we particularly thank Patrick Gill, Woodbury County Auditor and Recorder and Past President of the Iowa State Association of County Auditors; Elaine Johnston, County Auditor, Asotin County, Washington; Harvard L. Lomax, Registrar of Voters for Clark County, Nevada; Debbie Smith, Elections Coordinator, Caleveras County, California; Jocelyn Whitney, Developer and Project Manager for parallel testing activities in the State of California; Robert Williams, Chief Information Officer for Monmouth County, New Jersey; and Pam Woodside, former Chief Information Officer for the Maryland State Board of Elections. We would also like to acknowledge the National Committee for Voting Integrity for their cooperation and assistance in this effort.

Jeremy Creelan, Associate Attorney at Jenner & Block LLP, deserves credit for conceiving, launching, and supervising the Brennan Center's voting technology assessment project, including development of this report, as Deputy Director of the Center's Democracy Program through February 2005. The Program misses him greatly and wishes him well in private practice, where he continues to provide invaluable *pro bono* assistance.

The Brennan Center is grateful to Task Force member Lillie Coney, Associate Director of the Electronic Privacy Information Center. Among many other contributions, she provided invaluable assistance in assembling the Task Force, and frequently offered the Brennan Center sage strategic advice.

This report also benefited greatly from the insightful and thorough editorial assistance of Deborah Goldberg, Director of the Brennan Center's Democracy

Program. We are extremely grateful to Professor Henry Brady of the University of California at Berkeley and Professor Benjamin Highton of the University of California at Davis for their insights into the possible effects of denial of service attacks on voting systems. The Brennan Center also thanks Bonnie Blader, independent consultant, who provided the Task Force with crucial research, David M. Siegel, independent technology consultant, for his original contributions on the subject of software code inspections, and Tracey Lall, Ph.D. candidate in Computer Science at Rutgers University, who contributed many hours of critical security analysis. Douglas E. Dormer, CPA, CTP provided invaluable assistance in developing the analysis methodology and in keeping the task force focused. Joseph Lorenzo Hall also must be thanked for helping the Task Force members understand the diversity and commonality in voting system architectures. Much of the legal research was conducted by Gloria Garcia and Juan Martinez, J.D. candidates at Benjamin N. Cardozo School of Law, and Annie Lai and S. Michael Oliver, J.D. candidates at NYU School of Law. Lowell Bruce McCulley, CSSP, was exceptionally helpful in creating the attack catalogs. Finally, we thank Brennan Center Research Associates Annie Chen, Lauren Jones, Ana Munoz, and Neema Trivedi for their many hours of dedicated assistance.

Generous grants from an anonymous donor, the Carnegie Corporation of New York, the Ford Foundation, the HKH Foundation, the Knight Foundation, the Open Society Institute, and the Rockefeller Family Foundation supported the development and publication of this report. The statements made and views expressed in this report are the responsibility solely of the Brennan Center.

CONTENTS

Introduction . . . 1
Limitations of Study . . . 1
Summary of Findings and Recommendations . . . 3

The Need for a Methodical Threat Analysis . . . 6
Recurrent, Systematic Threat Analyses of Voting Systems Are Long Overdue . . . 6
Solid Threat Analyses Should Help Make Voting Systems More Reliable 6

Methodology . . . 8
Identification of Threats . . . 8
Prioritizing Threats – Number of Informed Participants as Metric . . . 8
Determining Number of Informed Participants . . . 10
Determining the Steps and Values for Each Attack . . . 10
Number of Informed Participants Needed to Change Statewide Election . . . 11
Limits of Informed Participants as Metric . . . 12
Effects of Implementing Countermeasure Sets . . . 13
Countermeasures Examined . . . 14
Basic Set of Countermeasures . . . 14
Inspection . . . 14
Physical Security for Machines . . . 14
Chain of Custody/Physical Security of Election Day Records . . . 15
Testing . . . 15
Regimen for Automatic Routine Audit Plus Basic Set of Countermeasures . . . 16
The Audit . . . 16
Transparent Random Selection Process . . . 17
Regimen for Parallel Testing Plus Basic Set of Countermeasures . . . 18
Parallel Testing . . . 18
Transparent Random Selection Process . . . 19

Representative Model for Evaluation of Attacks and Countermeasures: Governor's Race, State of Pennasota, 2007 . . . 20
Facts About Pennasota . . . 20
Evaluating Attacks in Pennasota . . . 20
Limits on Attacker . . . 22
Targeting the Fewest Counties . . . 23
Testing the Robustness of Our Findings . . . 23

The Catalogs . . . 24
Nine Categories of Attacks . . . 24

Lessons from the Catalogs: Retail Attacks Should Not Change the Outcome of Most Close Statewide Elections 27

Software Attacks on Voting Machines 30
History of Software-Based Attacks 30
Vendor Desire to Prevent Software Attack Programs 32
Inserting the Attack Program 33
Points of Attack: COTS and Vendor Software 33
Points of Attack: Software Patches and Updates 35
Points of Attack: Configuration Files and Election Definitions 35
Points of Attack: Network Communication .36
Points of Attack: Device Input/Output 36
Technical Knowledge 36
Election Knowledge 37
Attacking the Top of the Ticket 37
Parameterization 38
Creating an Attack Program That Changes Votes 39
Changing System Settings or Configuration Files 39
Active Tampering with User Interaction or Recording of Votes 40
Tampering with Electronic Memory After the Fact 40
Eluding Independent Testing Authority Inspections 42
Create Different Human-Readable and Binary Code 42
Use Attack Compiler, Linker, Loader or Firmware 42
Avoiding Inspection Altogether 43
Avoiding Detection During Testing 44
Avoiding Detection After the Polls Have Closed 44
Deciding How Many Votes to Change 45
Avoiding Event and Audit Logs 45
Coordinating with Paper Record Attacks 46
Conclusions 47

Least Difficult Attacks Applied Against Each System 48
Attacks Against DREs without VVPT 48
Representative “Least Difficult” Attack: Trojan Horse Inserted Into Operating System (DRE Attack No. 4) 49
Description of Potential Attack 49
How the Attack Could Swing Statewide Election 50
Effect of Basic Set of Countermeasures 51
Effect of Regimen for Parallel Testing 52
Infiltrating the Parallel Testing Teams 53
Creating an Attack That Recognizes Testing 53
Warning the Trojan Horse 54
Detecting the Test Environment 56
Recognizing Voting Patterns 57
Recognizing Usage Patterns 58

Taking Action When Parallel Testing Finds Discrepancies . .59
Conclusions and Observations 59
Attacks Against DREs w/ VVPT 61
Representative "Least Difficult" Attack: Trojan Horse Triggered with Hidden Commands in Ballot Definition File (DRE w/ VVPT Attack No. 1a) 62
Attacking Both Paper and Electronic Records (DRE w/ VVPT Attack No. 6) 65
Paper Misrecords Vote 65
Do Voters Review VVPT? 66
Effect of Regimen for Parallel Testing Plus Basic Set of Countermeasures 68
Effect of Regimen for Automatic Routine Audit Plus Basic Set of Countermeasures 68
Trojan Horse Attacks Paper at Time of Voting, Voters Fail to Review 69
Co-opting The Auditors 71
Replacing Paper Before the Automatic Routine Audit Takes Place 71
Replacing Some Paper Records Merely to Add Votes 73
Taking Action When Automatic Routine Audit Finds Anomalies 74
Conclusions 75
Attacks Against PCOS 77
Representative "Least Difficult" Attack: Software Attack Inserted on Memory Cards (PCOS Attack No. 41) 78
Description of Attack 78
Effect of Basic Set of Countermeasures .80
Effect of Regimen for Parallel Testing Plus Basic Set of Countermeasures 80
Effect of Regimen for Automatic Routine Audit Plus Basic Set of Countermeasures 81
PCOS Attack Number 42: Trojan Horse Disables Overvote Protections 81
PCOS Attack Number 49: Attack on Scanner Configuration Causes Misrecording of Votes 82
Conclusions 83

Prevention of Wireless Communication: A Powerful Countermeasure for All Three Systems .85

Security Recommendations 87

Directions for the Future 92
Witness and Cryptographic Systems 92
Informing Voters of Their Role in Making Systems More Secure 92

Additional Statistical Technical Techniques to Detect Fraud 92
Looking for Better Parallel Testing Techniques 93
Looking at Other Attack Goals 93
Looking at Other Races 93

Glossary 94

Appendices A-M 96

Endnotes 133

Figures

Figure 1. Voting Systems 2
Figure 2. Election for Governor, State of Pennasota, 2007 20
Figure 3. Assumed Precautions Taken by Attacker:
Limits on the % of Votes Added or Subtracted for a Candidate . .22
Figure 4. Total Votes Johnny Adams Needs to Switch
to Ensure Victory: 51,891 23
Figure 5. Typical Flow of Information To and From Voting Machines 24
Figure 6, Software Attack Program: Points of Entry
Figure 7. Possible Attack on DRE with VVPT 64
Figure 8. 66
Figure 9. 67
Figure 10. 70

INTRODUCTION

Problems with voting system security are making headlines like never before. The issue is attracting attention because of a number of factors: the rash of close, high-profile elections since 2000, greater attention to security since September 11, 2001, the recent shift in many states from mechanical to computerized voting systems, and high-profile reports about hacking of common electronic voting machines.

Public attention to voting system security has the potential to be a positive force. Unfortunately, too much of the public discussion surrounding security has been marred by claims and counter-claims that are based on little more than speculation or anecdote.

In response to this uninformed discussion, and with the intention of assisting election officials and the public as they make decisions about their voting machines, the Brennan Center for Justice at NYU School of Law assembled a Task Force of internationally renowned government, academic and private-sector scientists, voting machine experts, and security professionals to perform a methodical threat analysis of the voting systems most commonly purchased today. This is, as far as we know, the first systematic threat analysis of these voting systems. The methodology, analysis, and text were extensively peer reviewed by the National Institute of Standards and Technology ("NIST").

In this report, the Task Force reviews several categories of threats to the technologies of three electronic voting systems. Direct Recording Electronic voting systems ("DREs"), DREs with a voter-verified auditable paper trail ("DREs w/ VVPT") and Precinct Count Optical Scan ("PCOS") systems. We then identify, as against each system, the least difficult way for an attacker to change the outcome of a statewide election. And finally, we examine how much more difficult different sets of countermeasures would make these least difficult attacks. We believe that this analysis, together with the concurrent findings and recommended countermeasures, should assist jurisdictions decide which voting systems to certify or purchase, and how to protect those systems from security threats after they have been purchased.

This analysis should assist jurisdictions decide which voting systems to certify or purchase, and how to protect those systems from security threats after they have been purchased.

LIMITATIONS OF STUDY

As the first of its kind, this report is necessarily limited in scope. First, it is limited to voting systems that are being widely purchased *today*. The study does not include threat analyses of, most notably, ballot-marking devices,[1] vote by phone systems,[2] or ballot on demand, cryptographic, or witness voting systems.[3] Nor does this study consider early voting or voting that takes place through the mail.[4] We believe that the information and analysis included in this report can be used to perform threat analyses that include these systems and voting methods.

Second, our threat analysis is made in the context of a hypothetical statewide race. There is no reason why the methods used in this analysis cannot be applied to local (or national) races. We believe that such analyses would also be helpful in assisting jurisdictions with certification, purchase, and security decisions, but they were outside the scope of this study.

Third, our study is limited to an analysis of *technology-specific* threats. There are many types of potential attacks on election accuracy and credibility. We have not analyzed technology-neutral threats such as voter intimidation, illegal manipulation of voter rolls, or purges of voter rolls. We believe that such threats must be addressed. Because these threats are not specific to any particular voting system (*i.e.,* they should have the same impact on elections, regardless of the type of system a jurisdiction uses), however, they were not part of our study.

FIGURE 1

VOTING SYSTEMS

Type of Voting System	Description of Voting System	Examples of Voting System
Direct Recording Electronic (DRE)	A DRE machine directly records the voter's selections in each contest, using a ballot that appears on a display screen. Typical DRE machines have flat panel display screens with touch-screen input, although other display technologies have been used. The defining characteristic of these machines is that votes are captured and stored electronically.	Microvote Infinity Voting Panel Hart InterCivic eSlate Sequoia AVC Edge Sequoia AVC Advantage ES&S iVotronic ES&S iVotronic LS Diebold AccuVote-TS Diebold AccuVote-TSX Unilect Patriot
DRE with Voter Verified Paper Trail (DRE w/ VVPT)	A DRE w/ VVPT captures a voter's choice both internally in electronic form, and contemporaneously on paper. A DRE w/ VVPT allows the voter to confirm the accuracy of the paper record to provide voter-verification.	ES&S iVotronic system with Real Time Audit Log Diebold AccuVote-TSX with AccuView printer Sequoia AVC Edge with VeriVote printer Hart InterCivic eSlate with VVPAT Unilect Patriot with VVPAT
Precinct Count Optical Scan (PCOS)	PCOS voting machines allow voters to mark paper ballots, typically with pencils or pens, independent of any machine. Voters then carry their sleeved ballots to a scanner. At the scanner, they un-sleeve the ballot and insert into the scanner, which optically records the vote.	Diebold AccuVote-OS ES&S Model 100 Sequoia Optech Insight

Fourth, our analysis assumed that certain fundamental physical security and accounting procedures were already in place. Without good procedures, no voting system can be secured. We assumed the operation of a consistent set of procedures drawn from interviews with election officials in order to evaluate the number of informed participants involved in a given attack. All three systems are more vulnerable to attack if appropriate internal controls and procedures are not followed.

All three systems are more vulnerable to attack if appropriate internal controls and procedures are not followed.

Fifth, the report does not address other important factors that must be considered when making decisions about voting systems. Separate from (but concurrent with) its work with the Task Force on Voting System Security, the Brennan Center has completed a series of reports with task forces on voting system accessibility, usability, and cost.[5] In making decisions about their voting systems, jurisdictions must balance their security concerns with important concerns in these other areas.

Finally, our study looks at the ability of persons to successfully execute an attack without detection. Ultimately, it will be up to local jurisdictions to develop clear policies and procedures to ensure that when they find evidence of fraud or accident sufficient to change the outcome of a particular election, appropriate remedial action is taken.

SUMMARY OF FINDINGS AND RECOMMENDATIONS

Three fundamental points emerge from our threat analysis:

- All three voting systems have significant security and reliability vulnerabilities, which pose a real danger to the integrity of national, state, and local elections.
- The most troubling vulnerabilities of each system can be substantially remedied if proper countermeasures are implemented at the state and local level.
- Few jurisdictions have implemented any of the key countermeasures that could make the least difficult attacks against voting systems much more difficult to execute successfully.

Voting System Vulnerabilties

After a review of more than 120 potential threats to voting systems, the Task Force reached the following crucial conclusions:

For *all three* types of voting systems:

- When the goal is to change the outcome of a close statewide election, attacks that involve the insertion of Software Attack Programs or other corrupt software are the least difficult attacks.

- Voting machines that have wireless components are significantly more vulnerable to a wide array of attacks. Currently, only two states, New York and Minnesota, ban wireless components on all voting machines.

For *DREs without* voter-verified paper trails:

- DREs without voter-verified paper trails do not have available to them a powerful countermeasure to software attacks: post-election Automatic Routine Audits that compare paper records to electronic records.

For DREs w/ VVPT and PCOS:

- The voter-verified paper record, *by itself*, is of questionable security value. The paper record has significant value only if an Automatic Routine Audit is performed (and well designed chain of custody and physical security procedures are followed). Of the 26 states that mandate voter-verified paper records, only 12 require regular audits.

- Even if jurisdictions routinely conduct audits of voter-verified paper records, DREs w/ VVPT and PCOS are vulnerable to certain software attacks or errors. Jurisdictions that conduct audits of paper records should be aware of these potential problems.

Security Recommendations

There are a number of steps that jurisdictions can take to address the vulnerabilities identified in the threat analysis and thus to make their voting systems significantly more secure. Specifically, we recommend adoption of the following security measures: [6]

1. Conduct Automatic Routine Audits comparing voter-verified paper records to the electronic record following every election. A voter-verified paper record accompanied by a solid Automatic Routine Audit of those records can go a long way toward making the least difficult attacks much more difficult.

2. Perform "Parallel Testing" (selection of voting machines at random and testing them as realistically as possible) on Election Day. For paperless DREs, in particular, Parallel Testing will help jurisdictions detect software-based attacks as well as subtle software bugs that may not be discovered during inspection and other testing. The Task Force does not recommend Parallel Testing as a substitute for the use of voter verified paper records with an Automatic Routine Audit.

3. Ban use of voting machines with wireless components. All three voting systems are more vulnerable to attack if they have wireless components.

4. Use a transparent and random selection process for all auditing procedures. For any auditing to be effective (and to ensure that the public is confident in such procedures), jurisdictions must develop and implement transparent and random selection procedures.

5. Ensure decentralized Programming and Voting System administration. Where a single entity, such as a vendor or state or national consultant, performs key tasks for multiple jurisdictions, attacks against statewide elections become easier.

6. Institute clear and effective procedures for addressing evidence of fraud or error. Both Automatic Routine Audits and Parallel Testing are of questionable security value without effective procedures for action where evidence of machine malfunction or fraud is discovered. Detection of fraud without an appropriate response will not prevent attacks from succeeding.

Fortunately, these steps are not particularly complicated or cumbersome. For the most part, they do not involve significant changes in system architecture. Unfortunately, *few jurisdictions have implemented any of the recommended countermeasures.*

Regular examinations of voting system security are necessary because we have *not* always successfully avoided attacks on voting systems

THE NEED FOR A METHODICAL THREAT ANALYSIS

Is an independent study of voting system security really necessary? Have we not managed, in our nation's 230 year history, to avoid the kind of attacks about which certain advocates are suddenly warning?

RECURRENT, SYSTEMATIC THREAT ANALYSES OF VOTING SYSTEMS ARE LONG OVERDUE.

The simple answer is that regular examinations of voting system security are necessary because we have *not* always successfully avoided attacks on voting systems – in fact, various types of attacks on voting systems and elections have a "long tradition" in American history.[7] The suspicion or discovery of such attacks has generally provoked momentary outrage, followed by periods of historical amnesia.[8]

In his 1934 book on this issue, Joseph Harris documented numerous cases of attacks on voting systems, including ballot box stuffing, alteration of ballots, substitution of ballots, false counts, posting of false returns, and alteration of returns.[9] More recent examples of tampering with voting systems have been exposed in the last two decades.[10]

In the past, when security and reliability issues surrounding elections have bubbled to the surface of public consciousness, Americans have embraced new technology.[11] It is therefore not particularly surprising that, following the controversial 2000 presidential elections, we have again turned to new voting machines to address our concerns.

These new machines promise great advancements in the areas of accessibility and usability. But all technology, no matter how advanced, is going to be vulnerable to attack to some degree. Many of the vulnerabilities present in our new voting technologies are the same that have always existed; some are new.

The main lesson of the history of attacks on voting systems is that we would be foolish to assume there would not be attacks on voting systems in the future. The best that we can do is understand what vulnerabilities exist, and take the proper precautions to ensure that the easiest attacks, with the potential to affect the most votes, are made as difficult as possible.

SOLID THREAT ANALYSES SHOULD HELP MAKE VOTING SYSTEMS MORE RELIABLE.

There is an additional benefit to this kind of analysis: it should help make our voting systems more reliable, *regardless of whether they are ever attacked*. Computerized voting systems – like all previous voting systems – have shown themselves vulnerable to error. Votes have been miscounted or lost as a result of defective firmware,[12] faulty machine software,[13] defective tally server software,[14] election programming errors,[15] machine breakdowns,[16] malfunctioning input devices,[17] and pollworker error.[18]

As Professor Douglas Jones has noted: "An old maxim in the area of computer security is clearly applicable here: Almost everything that a malicious attacker could attempt could also happen by accident; for every malicious attacker, there may be thousands of people making ordinary careless errors."[19] Solid threat analyses should help to expose and to address vulnerabilities in voting systems, not just to security breaches, but also to simple malfunctions that could be avoided.

The main lesson of the history of attacks on voting systems is that we would be foolish to assume there would not be attacks on voting systems in the future.

Firmware is software that is embedded in the voting machine.

METHODOLOGY

The Task Force concluded, and the peer review team at NIST agreed, that the best approach for comprehensively evaluating voting system threats was to: (1) identify and categorize the potential threats against voting systems, (2) prioritize these threats based upon an agreed upon metric (which would tell us how difficult each threat is to accomplish from the attacker's point of view), and (3) determine, utilizing the same metric employed to prioritize threats, how much more difficult each of the catalogued attacks would become after various sets of countermeasures are implemented.

This model allows us to identify the attacks we should be most concerned about (*i.e.,* the most practical and least difficult attacks). Furthermore, it allows us to quantify the potential effectiveness of various sets of countermeasures (*i.e.,* how difficult the least difficult attack is after the countermeasure has been implemented). Other potential models considered, but ultimately rejected by the Task Force, are detailed in Appendix B.

IDENTIFICATION OF THREATS

The first step in creating a threat model for voting systems was to identify as many potential attacks as possible. To that end, the Task Force, together with the participating election officials, spent several months identifying voting system vulnerabilities. Following this work, NIST held a Voting Systems Threat Analysis Workshop on October 7, 2005. Members of the public were invited to write up and post additional potential attacks. Taken together, this work produced over 120 potential attacks on the three voting systems. They are detailed in the catalogs annexed.[20] Many of the attacks are described in more detail at http://vote.nist.gov/threats/papers.htm.

The types of threats detailed in the catalogs can be broken down into nine categories: (1) the insertion of corrupt software into machines prior to Election Day; (2) wireless and other remote control attacks on voting machines on Election Day; (3) attacks on tally servers; (4) miscalibration of voting machines; (5) shut off of voting machine features intended to assist voters; (6) denial of service attacks; (7) actions by corrupt poll workers or others at the polling place to affect votes cast; (8) vote buying schemes; (9) attacks on ballots or VVPT. Often, the actual attacks involve some combination of these categories. We provide a discussion of each type of attack in "Categories of Attacks," *infra* at pp. 24–27.

PRIORITIZING THREATS: NUMBER OF INFORMED PARTICIPANTS AS METRIC

Without some form of prioritization, a compilation of the threats is of limited value. Only by prioritizing these various threats could we help election officials identify which attacks they should be most concerned about, and what steps

Only by prioritizing these various threats could we help election officials identify which attacks they should be most concerned about, and what steps could be taken to make such attacks as difficult as possible.

could be taken to make such attacks as difficult as possible. As discussed below, we have determined the level of difficulty for each attack where the attacker is attempting to affect the outcome of a close statewide election.[21]

There is no perfect way to determine which attacks are the least difficult, because each attack requires a different mix of resources – well-placed insiders, money, programming skills, security expertise, *etc.* Different attackers would find certain resources easier to acquire than others. For example, election fraud committed by local election officials would always involve well-placed insiders and a thorough understanding of election procedures; at the same time, there is no reason to expect such officials to have highly skilled hackers or first-rate programmers working with them. By contrast, election fraud carried out by a foreign government would likely start with plenty of money and technically skilled attackers, but probably without many conveniently placed insiders or detailed knowledge of election procedures.

Ultimately, we decided to use the "number of informed participants" as the metric for determining attack difficulty. An attack which uses fewer participants is deemed the easier attack.

We have defined "informed participant" as someone whose participation is needed to make the attack work, and who knows enough about the attack to foil or expose it. This is to be distinguished from a participant who unknowingly assists the attack by performing a task that is integral to the attack's successful execution without understanding that the task is part of an attack on voting systems.

The reason for using the security metric "number of informed participants" is relatively straightforward: the larger a conspiracy is, the more difficult it would be to keep it secret. Where an attacker can carry out an attack by herself, she need only trust herself. On the other hand, a conspiracy that requires thousands of people to take part (like a vote-buying scheme) also requires thousands of people to keep quiet. The larger the number of people involved, the greater the likelihood that one of them (or one who was approached, but declined to take part) would either inform the public or authorities about the attack, or commit some kind of error that causes the attack to fail or become known.

Moreover, recruiting a large number of people who are willing to undermine the integrity of a statewide election is also presumably difficult. It is not hard to imagine two or three people agreeing to work to change the outcome of an election. It seems far less likely that an attacker could identify and employ hundreds or thousands of similarly corrupt people without being discovered.

We can get an idea of how this metric works by looking at one of the threats listed in our catalogs: the vote-buying threat, where an attacker or attackers pay individuals to vote for a particular candidate. This is Attack Number 26 in the PCOS Attack Catalog[22] (though this attack would not be substantially different against

While practical in smaller contests, a vote-buying attack would be an exceptionally difficult way to affect the outcome of a statewide election.

DREs or DREs w/ VVPT).[23] In order to work under our current types of voting systems, this attack requires (1) at least one person to purchase votes, (2) many people to agree to sell their votes, and (3) some way for the purchaser to confirm that the voters she pays actually voted for the candidate she supported. Ultimately, we determined that, while practical in smaller contests, a vote-buying attack would be an exceptionally difficult way to affect the outcome of a statewide election. This is because, even in a typically close statewide election, an attacker would need to involve thousands of voters to ensure that she could affect the outcome of a statewide race.[24]

For a discussion of other metrics we considered, but ultimately rejected, *see* Appendix C.

DETERMINING NUMBER OF INFORMED PARTICIPANTS

DETERMINING THE STEPS AND VALUES FOR EACH ATTACK

The Task Force members broke down each of the catalogued attacks into its necessary steps. For instance, Attack 12 in the PCOS Attack Catalog is "Stuffing Ballot Box with Additional Marked Ballots."[25] We determined that, at a minimum, there were three component parts to this attack: (1) stealing or creating the ballots and then marking them, (2) scanning marked ballots through the PCOS scanners, probably before the polls opened, and (3) modifying the poll books in each location to ensure that the total number of votes in the ballot boxes was not greater than the number of voters who signed in at the polling place.

Task Force members then assigned a value representing the minimum number of persons they believed would be necessary to accomplish each goal. For PCOS Attack 12, the following values were assigned:[26]

Minimum number required to steal or create ballots: 5 persons total.[27]

Minimum number required to scan marked ballots: 1 per polling place attacked.

Minimum number required to modify poll books: 1 per polling place attacked.[28]

After these values were assigned, the Brennan Center interviewed several election officials to see whether they agreed with the steps and values assigned to each attack.[29] When necessary, the values and steps were modified. The new catalogs, including attack steps and values, were then reviewed by Task Force members. The purpose of this review was to ensure, among other things, that the steps and values were sound.

These steps and values tell us how difficult it would be to accomplish a *single attack in a single polling place.* They do not tell us how many people it would take to change the outcome of an election successfully – that depends, of course, on specific facts about the jurisdiction: how many votes are generally recorded in each polling place, how many polling places are there in the jurisdiction, and how close is the race? For this reason, we determined that it was necessary to construct a hypothetical jurisdiction, to which we now turn.

NUMBER OF INFORMED PARTICIPANTS NEEDED TO CHANGE STATEWIDE ELECTION

We have decided to examine the difficulty of each attack in the context of changing the outcome of a reasonably close statewide election. While we are concerned by potential attacks on voting systems in any type of election, we are most troubled by attacks that have the potential to affect large numbers of votes. These are the attacks that could actually change the outcome of a statewide election with just a handful of attack participants.

We are less troubled by attacks on voting systems that can only affect a small number of votes (and might therefore be more useful in local elections). This is because there are many non-system attacks that can also affect a small number of votes (*i.e.*, sending out misleading information about polling places, physically intimidating voters, submitting multiple absentee ballots, *etc.*). Given the fact that these non-system attacks are likely to be less difficult in terms of number of participants, financial cost, risk of detection, and time commitment, we are uncertain that an attacker would target *voting machines* to alter a small number of votes.

In order to evaluate how difficult it would be for an attacker to change the outcome of a statewide election, we created a composite jurisdiction. The composite jurisdiction was created to be representative of a relatively close statewide election. We did not want to examine a statewide election where results were so skewed toward one candidate (for instance, the re-election of Senator Edward M. Kennedy in 2000, where he won 73% of the vote[30]), that reversing the election results would be impossible without causing extreme public suspicion. Nor did we want to look at races where changing only a relative handful of votes (for instance, the Governor's race in Washington State in 2004, which was decided by a mere 129 votes[31]) could affect the outcome of an election; under this scenario, many of the potential attacks would involve few people, and therefore look equally difficult.

We have named our composite jurisdiction "the State of Pennasota." The State of Pennasota is a composite of ten states: Colorado, Florida, Iowa, Ohio, New Mexico, Pennsylvania, Michigan, Nevada, Wisconsin and Minnesota. These states were chosen because they were the ten "battleground" states that Zogby International consistently polled in the spring, summer, and fall 2004.[32] These are statewide elections that an attacker would have expected, ahead of time, to be fairly close.

We have also created a composite election, which we label the "Governor's Race" in Pennasota. The results of this election are a composite of the actual results in the same ten states in the 2004 Presidential Election.

We have used these composites as the framework by which to evaluate the difficulty of the various catalogued attacks.[33] For instance, we know a ballot-box stuffing attack would require roughly five people to create and mark fake ballots, as well as one person per polling place to stuff the boxes, and one person per polling place to modify the poll books. But, in order to determine how many informed participants would be needed to affect a statewide race, we need to know how many polling places would need to be attacked.

The composite jurisdiction and composite election provide us with information needed to answer these questions: *i.e.*, how many extra votes our attackers would need to add to their favored candidate's total for him to win, how many ballots our attackers can stuff into a particular polling place's ballot box without arousing suspicion (and related to this, how many votes are generally cast in the average polling place), how many polling places are there in the state, *etc.* We provide details about both the composite jurisdiction and election in the section entitled "Governor's Race, State of Pennasota, 2007," *infra* at pp 20–27.

LIMITS OF INFORMED PARTICIPANTS AS METRIC

Of the possible metrics we considered, we believe that measuring the number of people who know they are involved in an attack (and thus could provide evidence of the attack to the authorities and/or the media), is the best single measure of attack difficulty; as already discussed, we have concluded that the more people an attacker is forced to involve in his attack, the more likely it is that one of the participants would reveal the attack's existence and foil the attack, perhaps sending attackers to jail. However, we are aware of a number of places where the methodology could provide us with questionable results.

By deciding to concentrate on size of attack team, we mostly ignore the need for other resources when planning an attack. Thus, a software attack on DREs which makes use of steganography[34] to hide attack instruction files (*see* "DRE w/ VVPT Attack No.1a", discussed in greater detail, *infra* at pp. 62–65) is considered easier than an attack program delivered over a wireless network at the polling place (*see* discussion of wireless networks, *infra* at pp. 85–91). However, the former attack probably requires a much more technologically sophisticated attacker.

Another imperfection with this metric is that we do not have an easy way to represent how much choice the attacker has in finding members of his attack team. Thus, with PCOS voting, we conclude that the cost of subverting a routine audit of ballots is roughly equal to the cost of intercepting ballot boxes in transit and substituting altered ballots (*see* discussion of PCOS attacks, *infra* at pp. 77–83). However, subverting the audit team requires getting a specific set of trusted people to cooperate with the attacker. By contrast, the attacker may be able to decide which precincts to tamper with based on which people he has already recruited for his attack.

In an attempt to address this concern, we considered looking at the number of "insiders" necessary to take part in each attack. Under this theory, getting five people to take part in a conspiracy to attack a voting system might not be particularly difficult. But getting five well-placed county election officials to take part in the attack would be (and should be labeled) the more difficult of the two attacks. Because, for the most part, the low-cost attacks we have identified do not necessarily involve well placed insiders (but could, for instance, involve one of many people with access to commercial off the shelf software ("COTS") during development or at the vendor), we do not believe that using this metric would have substantially changed our analysis.[35]

Finally, these attack team sizes do not always capture the logistical complexity of an attack. For example, an attack on VVPT machines involving tampering with the voting machine software and also replacing the paper records in transit requires the attacker to determine what votes were falsely produced by the voting machine and print replacement records in time to substitute them. While this is clearly possible, it raises a lot of operational difficulties – a single failed substitution leaves the possibility that the attack would be detected during the audit of ballots.

We have tried to keep these imperfections in mind when analyzing and discussing our least difficult attacks.

We suspect that much of the disagreement between voting officials and computer security experts in the last several years stems from a difference of opinion in prioritizing the difficulty of attacks. Election officials, with extensive experience in the logistics of handling tons of paper ballots, have little faith in paper and understand the kind of breakdowns in procedures that lead to traditional attacks like ballot box stuffing; in contrast, sophisticated attacks on computer voting systems appear very difficult to many of them. Computer security experts understand sophisticated attacks on computer systems, and recognize the availability of tools and expertise that makes these attacks practical to launch, but have no clear idea how they would manage the logistics of attacking a paper-based system. Looking at attack team size is one way to bridge this difference in perspective.

■ EFFECTS OF IMPLEMENTING COUNTERMEASURE SETS

The final step of our threat analysis is to measure the effect of certain countermeasures against the catalogued attacks. How much more difficult would the attacks become once the countermeasures are put into effect? How many more informed participants (if any) would be needed to counter or defeat these countermeasures?

Our process for examining the effectiveness of a countermeasure mirrors the process for determining the difficulty of an attack: we first asked whether the countermeasure would allow us to detect an attack with near certainty. If we agreed that the countermeasure would expose the attack, we identified the steps that would be necessary to circumvent or defeat the countermeasure. For each step to defeat the countermeasure, we determined the number of additional informed participants (if any) that an attacker would need to add to his team.

As with the process for determining attack difficulty, the Brennan Center interviewed numerous election officials to see whether they agreed with the steps and values assigned. When necessary, the values and steps for defeating the countermeasures were altered to reflect the input of election officials.

COUNTERMEASURES EXAMINED

BASIC SET OF COUNTERMEASURES

The first set of countermeasures we looked at is the "Basic Set" of countermeasures. This Basic Set was derived from security survey responses[36] we received from county election officials around the country, as well as additional interviews with more than a dozen current and former election officials. Within the Basic Set of countermeasures are the following procedures:

Inspection

- The jurisdiction is not knowingly using any uncertified software that is subject to inspection by the Independent Testing Authority (often referred to as the "ITA").[37]

Physical Security for Machines

- Ballot boxes (to the extent they exist) are examined (to ensure they are empty) and locked by poll workers immediately before the polls are opened.

- Before and after being brought to the polls for Election Day, voting systems for each county are locked in a single room, in a county warehouse.

- The warehouse has perimeter alarms, secure locks, video surveillance and regular visits by security guards.

- Access to the warehouse is controlled by sign-in, possibly with card keys or similar automatic logging of entry and exit for regular staff.

- Some form of "tamper evident" seals are placed on machines before and after each election.

- The machines are transported to polling locations five to fifteen days before Election Day.

Chain of Custody/Physical Security of Election Day Records

- At close of the polls, vote tallies for each machine are totaled and compared with number of persons that have signed the poll books.

- A copy of totals for each machine is posted at each polling place on Election Night and taken home by poll workers to check against what is posted publicly at election headquarters, on the web, in the papers, or elsewhere.[38]

- All audit information (*i.e.*, Event Logs, VVPT records, paper ballots, machine printouts of totals) that is not electronically transmitted as part of the unofficial upload to the central election office, is delivered in official, sealed and hand-delivered information packets or boxes. All seals are numbered and tamper-evident.

- Transportation of information packets is completed by two election officials representing opposing parties who have been instructed to remain in joint custody of the information packets or boxes from the moment it leaves the precinct to the moment it arrives at the county election center.

- Each polling place sends its information packets or boxes to the county election center separately, rather than having one truck or person pick up this data from multiple polling locations.

- Once the sealed information packets or boxes have reached the county election center, they are logged. Numbers on the seals are checked to ensure that they have not been replaced. Any broken or replaced seals are logged. Intact seals are left intact.

- After the packets and/or boxes have been logged, they are provided with physical security precautions at least as great as those listed for voting machines, above. Specifically, for Pennasota, we have assumed the room in which the packets are stored have perimeter alarms, secure locks, video surveillance and regular visits by security guards and county police officers; and access to the room is controlled by sign-in, possibly with card keys or similar automatic logging of entry and exit for regular staff.

Testing[39]

- An Independent Testing Authority has certified the model of voting machine used in the polling place.

- Acceptance Testing[40] is performed on machines at time, or soon after they are received by County.

- Pre-election Logic and Accuracy[41] testing is performed by the relevant election official.

- Prior to opening the polls, every voting machine and vote tabulation system is checked to see that it is still configured for the correct election, including the correct precinct, ballot style, and other applicable details.

REGIMEN FOR AUTOMATIC ROUTINE AUDIT PLUS BASIC SET OF COUNTERMEASURES.

The second set of countermeasures is the Regimen for an Automatic Routine Audit Plus Basic Set of Countermeasures.

Some form of routine auditing of voter-verified paper records occurs in 12 states, to test the accuracy of electronic voting machines. They generally require between 1 and 10% of all precinct voting machines to be audited after each election.[42]

Jurisdictions can implement this set of countermeasures only if their voting systems produce some sort of voter-verified paper record of each vote. This could be in the form of a paper ballot, in the case of PCOS, or a voter-verified paper trail ("VVPT"), in the case of DREs.

We have assumed that jurisdictions take the following steps when conducting an Automatic Routine Audit (when referring to this set of assumptions "Regimen for an Automatic Routine Audit"):

The Audit

- Leaders of the major parties in each county are responsible for selecting a sufficient number of audit-team members to be used in that county.[43]

- Using a highly transparent random selection mechanism (*see* point ii, below), the voter-verified paper records for between a small percentage of all voting machines in the State are selected for auditing.

- Using a transparent random selection method, auditors are assigned to the selected machines (two or three people, with representatives of each major political party, would comprise each audit team).

- The selection of voting machines, and the assignment of auditors to machines, occurs immediately before the audits take place. The audits take place as soon after polls close as possible – for example, at 9 a.m. the morning after polls close.

- Using a transparent random selection method, county police officers, security personnel and the video monitor assigned to guard the voter-verified records are chosen from a large pool of on-duty officers and employees on election night.
- The auditors are provided the machine tallies and are able to see that the county tally reflects the sums of the machine tallies before the start of the inspection of the paper.
- The audit would include a tally of spoiled ballots (in the case of VVPT, the number of cancellations recorded), overvotes, and undervotes.

Transparent Random Selection Process

In this report, we have assumed that random auditing procedures are in place for both the Regimen for an Automatic Routine Audit and Regimen for Parallel Testing. We have further assumed procedures to prevent a single, corrupt person from being able to fix the results. This implies a kind of transparent and public random procedure.

For the Regimen for an Automatic Routine Audit there are at least two places where transparent, random selection processes are important: in the selection of precincts to audit, and in the assignment of auditors to the precincts they will be auditing.

Good election security can employ Transparent Random Selection in other places with good effect:

- the selection of parallel testers from a pool of qualified individuals.
- the assignment of police and other security professionals from on-duty lists, to monitor key materials, for example, the VVPT records between the time that they arrive at election central and the time of the completion of the ARA.

If a selection process for auditing is to be trustworthy and trusted, ideally:

- The whole process will be publicly observable or videotaped;[44]
- The random selection will be publicly verifiable, *i.e.,* anyone observing will be able to verify that the sample was chosen randomly (or at least that the number selected is not under the control of any small number of people); and

- The process will be simple and practical within the context of current election practice so as to avoid imposing unnecessary burdens on election officials.

There are a number of ways that election officials can ensure some kind of transparent randomness. One way would be to use a state lottery machine to select precincts or polling places for auditing. We have included two potential examples of transparent random selection processes in Appendix F. These apply to the Regimen for Parallel Testing as well.

REGIMEN FOR PARALLEL TESTING PLUS BASIC SET OF COUNTERMEASURES

The final set of countermeasures we have examined is "Parallel Testing" plus the Basic Set of countermeasures. Parallel Testing, also known as election-day testing, involves selecting voting machines at random and testing them as realistically as possible during the period that votes are being cast.

Parallel Testing

In developing our set of assumptions for Parallel Testing, we relied heavily upon interviews with Jocelyn Whitney, Project Manager for Parallel Testing in the State of California, and conclusions drawn from this Report.[45] In our analysis, we assume that the following procedures would be included in the Parallel Testing regimen (when referring to this regimen "Regimen for Parallel Testing") that we evaluate:

- At least two of each DRE model (meaning both vendor and model) would be selected for Parallel Testing;

- At least two DREs from each of the three largest counties would be parallel tested;

- Counties to be parallel tested would be chosen by the Secretary of State in a transparent and random manner.

- Counties would be notified as late as possible that machines from one of their precincts would be selected for Parallel Testing;[46]

- Precincts would be selected through a transparent random mechanism;

- A video camera would record testing;

- For each test, there would be one tester and one observer;

- Parallel Testing would occur at the polling place;

- The script for Parallel Testing would be generated in a way that mimics voter behavior and voting patterns for the polling place;

- At the end of the Parallel Testing, the tester and observer would reconcile vote totals in the script with vote totals reported on the machine.

Transparent Random Selection Process

We further assume that the same type of transparent random selection process that would be used for the Regimen for Automatic Routine Audit would also be employed for the Regimen for Parallel Testing to determine which machines would be subjected to testing on Election Day.

REPRESENTATIVE MODEL FOR EVALUATING ATTACKS AND COUNTERMEASURES: GOVERNOR'S RACE, STATE OF PENNASOTA, 2007

In this section, we provide the assumptions that we have made concerning (1) the governor's race in the State of Pennasota, and (2) the limitations that our attacker would face in attempting to subvert that election.

FACTS ABOUT PENNASOTA

In creating our assumptions for the Pennasota's gubernatorial race, we have averaged the results of the 2004 Presidential Election in ten "battleground" states. Based upon this average, we have assumed that 3,459,379 votes would be cast in Pennasota's gubernatorial election. The average margin of victory in the 10 battleground states was 2.3%. Accordingly, we assumed that this would be the margin of victory between the two main candidates in our hypothetical election (in total votes, this is 80,257).

FIGURE 2

ELECTION FOR GOVERNOR, STATE OF PENNASOTA, 2007

Candidate	Party	Total Votes	Percentage of Votes
Tom Jefferson	Dem-Rep	1,769,818	51.1
Johnny Adams	Federalists	1,689,650	48.8

A table that documents all of the relevant numbers for Pennasota and the 2007 gubernatorial election is provided in Appendix G.[49]

EVALUATING ATTACKS IN PENNASOTA

To complete our analysis, we ran each attack through the 2007 governor's race in Pennasota. The goal was to determine how many informed participants would be needed to move the election from Tom Jefferson to Johnny Adams.

We have assumed that our attacker would seek to change these results so that Johnny Adams is assured victory. Accordingly, although the election is decided by 2.3% of the vote, we have calculated that the attacker's goal is to (1) add 3.0% (or 103,781 votes) to Johnny Adams total, (2) subtract 3.0% of the total votes from Tom Jefferson, or (3) switch 1.5% (or 51,891 votes) from Tom Jefferson to Johnny Adams. [50]

By examining a particular attack in the context of our goal of changing the results of Pennasota's 2007 governor's race, it becomes clear how difficult an attack actually would be. Earlier, we assigned the following steps and values for PCOS Attack 12 ("Stuffing Ballot Box with Additional Marked Ballots"):

Minimum number required to steal or create ballots:[51] 5 persons total

Minimum number required to scan the ballots: 1 per polling place attacked.

Minimum number required to modify poll books: 1 per polling place attacked.

Our attacker seeks to use the "ballot stuffing attack" to add 103,781 votes to Johnny Adams' total. There are approximately 1142 voters per polling place in the State of Pennasota.[52] Theoretically, our attacker could add 103,781 votes for Johnny Adams in the boxes of three or four polling places and her favored candidate would win. In this case, she would only need to involve a dozen people (including herself) to carry out the attack successfully: five to create the ballots, three or four to stuff the boxes, and three or four to modify (and add to) the poll books.

As a practical matter, of course, this attempt at ballot stuffing would not work. Someone (and, more likely, many people) would notice if a few polling places that normally recorded 1100-1200 votes were suddenly reporting 25,000 votes each for Johnny Adams.

We have assumed that, in order to avoid detection, our attacker could add no more than 15% of the total votes in a particular polling place for Johnny Adams (*see* "Limits on Attacker," *infra* at p. 12, for further discussion). Accordingly, our formula for determining how many polling places she must target is as follows:

> Polling places targeted = total votes that must be added/(votes per polling place x percent that may be taken from any polling place)

or, in actual numbers:

> Polling places targeted = 103,781/(1,142 x 15%) = 606

From this we learn that attempting to change a statewide election by scanning in extra marked ballots would be extremely difficult. More specifically, it would likely require more than 1,000 informed participants: 5 to create/steal and mark the appropriate ballots, plus 606 to place in separate ballot boxes in each polling place, plus 606 to modify the poll books in each polling place. It is unlikely that (1) an attacker could find so many people willing to participate in such an attack without inadvertently soliciting someone who would expose the plot, (2) all 1,000 participants would keep silent about the attack, and (3) even if all 1,000 solicited persons agreed to take part in the attack, and none of them purposefully exposed the plot, that no one would get caught perpetrating the conspiracy.[53]

LIMITS ON ATTACKER

We have assumed that our attacker would prefer that her actions not raise undue suspicion. Accordingly, we have placed some limits on the type of actions our attacker could take. As just demonstrated by looking at the ballot stuffing attack, these limits can further help us determine how difficult a particular attack would be (*i.e.*, how many informed participants the attacker would need to involve).

Perhaps most importantly, we have assumed our attacker would not want to add or subtract more than 10% of the votes for a candidate in any one county (or switch more than 5% from one candidate to another), for fear that a greater change would attract suspicion. We believe that this is a conservative estimate, but the reason for creating some kind of cap should be obvious: if enough votes are switched in a specific location, it would eventually become apparent that something has gone wrong (whether through fraud or error).

We can see this by looking at a specific example from an actual election. In 2004, in heavily Democratic Cook County, Illinois, John Kerry received 59% of the vote and George Bush received 40%.[54] It is unlikely that, just by looking at vote totals for Cook County, anyone would have assumed that there was fraud or error if John Kerry received 63% or 55% of the countywide vote. On the other hand, if John Kerry received less than 50% or more than 70% of the vote in Cook County, these totals would (at the very least) attract attention and increase the likelihood that there would be some investigation. This would be particularly true if John Kerry's totals were otherwise within reasonable expectations in other counties in Illinois and around the country. An attacker would seek to avoid such an extraordinary aberration.

For the same reasons, we have put limits on the number of votes an attacker would seek to change in a single polling place or a single machine. We have assumed that a swing of greater than 15% in any single polling place or 30% on any single machine would attract too much suspicion, and that, therefore, an attacker would avoid adding or subtracting more than these numbers of votes per polling place and machine.[55]

FIGURE 3

ASSUMED PRECAUTIONS TAKEN BY ATTACKER:
LIMITS ON THE % OF VOTES ADDED OR SUBTRACTED FOR A CANDIDATE

Maximum % Votes Added or Subtracted Per County	10% (5% switch)
Maximum % Votes Added or Subtracted Per Polling Place	15% (7.5% switch)
Maximum % Votes Added or Subtracted Per Voting Machine	30% (15% switch)

TARGETING THE FEWEST COUNTIES

As will be discussed, *infra* at pp. 71–74, many attacks would be easier to execute, and more difficult to detect, if they were limited to a small number of counties or polling places. Given the limits we have set on our attacker, we have concluded that, to change enough votes to affect the outcome of our statewide election, she would have to attack a minimum of three counties.[56] These would be the three largest counties in the State of Pennasota (where there are enough votes to swing the statewide election).[57] This conclusion is supported in the table below.

We ran our threat analysis against the results of the 2004 presidential race in Florida, New Mexico and Pennsylvania.

FIGURE 4

TOTAL VOTES JOHNNY ADAMS NEEDS TO SWITCH TO ENSURE VICTORY: 51,891

	Actual Vote	Number of Votes Switched	% of County Votes Switched	New Total
Mega County		23,453	4.4%	
Jefferson (D-R)	194,848			171,395
Adams (F)	336,735			360,188
Capitol County		17,306	4.8%	
Jefferson (D-R)	157,985			140,679
Adams (F)	202,556			219,862
Suburbia County		11,132	4.2%	
Jefferson (D-R)	128,933			117,801
Adams (F)	135,003			146,135
Statewide Totals		51,891		
Jefferson (D-R)	1,769,818			1,717,927
Adams (F)	1,689,561			1,741,452

TESTING THE ROBUSTNESS OF OUR FINDINGS

To ensure that the results of our analysis were robust and not limited to the composite jurisdiction of Pennasota, we ran our threat analysis against the results of the 2004 presidential race in Florida, New Mexico and Pensylvania, and came up with substantially similar conclusions. Specifically, all of the findings and recommendations in the Introduction (*supra* at pp. 1–5) still applied.

We also re-ran our analysis in Pennasota, but changed the limits on our attacker, allowing her to change many more votes on a single machine and attempt to change the governor's race in a single (*i.e.*, "Mega") county. Again, all eight of the findings listed in the Executive Summary still applied.

THE CATALOGS

As already discussed, we have catalogued over 120 potential attacks on voting systems. These fall into nine categories, which cover the diversity and breadth of voting machine vulnerabilities.[59]

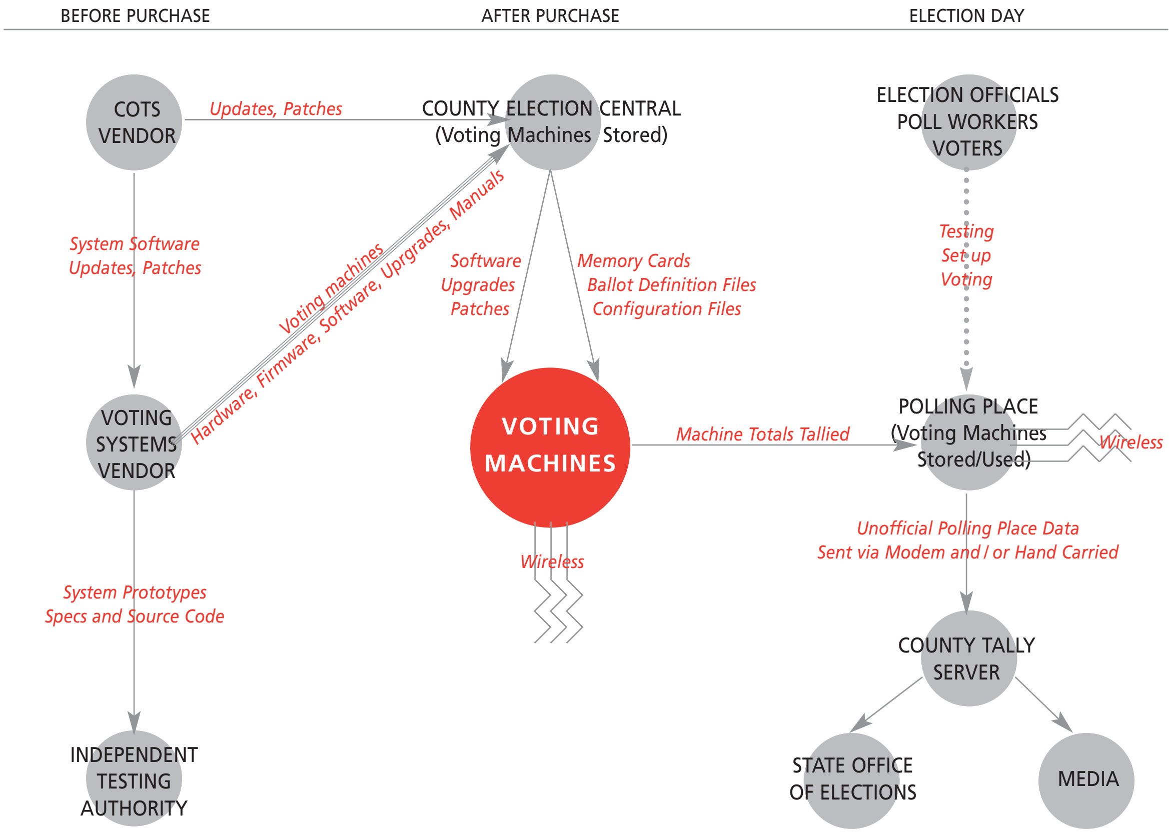
NINE CATEGORIES OF ATTACKS

One way of thinking about the voting process is to view it as a flow of information: the vendor and programmers present the voter with information about her election choices via the voting machine; the voter provides the voting machine with her choices; the voter's choice is then tallied by the voting machines, and this tallied information is (at the close of the polls) provided to poll workers; from the polling place, the vote tallies (whether in paper, electronic, or both forms) from all voting machines are sent to a county tally center; from there countywide totals are reported to state election officials and the media.

Attacks on voting systems are attacks on this flow of information. If we view the nine categories in the context of this flow, we get a better idea of how they might be accomplished.

FIGURE 5

TYPICAL FLOW OF INFORMATION TO AND FROM VOTING MACHINES


BEFORE PURCHASE
AFTER PURCHASE
ELECTION DAY
COTS VENDOR
Updates, Patches
COUNTY ELECTION CENTRAL (Voting Machines Stored)
ELECTION OFFICIALS
POLL WORKERS
VOTERS
System Software Updates, Patches
Voting machines
Hardware, Firmware, Software, Upgrades, Manuals
Software Upgrades Patches
Memory Cards
Ballot Definition Files
Configuration Files
Testing
Set up
Voting
VOTING SYSTEMS VENDOR
VOTING MACHINES
Machine Totals Tallied
POLLING PLACE (Voting Machines Stored/Used)
Wireless
Wireless
Unofficial Polling Place Data Sent via Modem and / or Hand Carried
System Prototypes Specs and Source Code
COUNTY TALLY SERVER
INDEPENDENT TESTING AUTHORITY
STATE OFFICE OF ELECTIONS
MEDIA

1. The Insertion of Corrupt Software Into Machines Prior to Election Day. This is an attack on the voting machine itself, and it occurs before the voting machine even reaches the polling place. Someone with access to voting machines, software, software updates, or devices inserted into voting machines (such as printers or memory cards) introduces corrupt software (such as an Attack Program) that forces the machine to malfunction in some way. We can see by looking at the chart that there are several points of attack that exist before a machine reaches the polling place. The malfunction triggered by the corrupt software could, among other things, cause the machine to misrecord votes, add or lose votes, skip races, perform more slowly or break down altogether.

One challenge associated with this attack is that it is likely to be operationally and technically difficult to carry out successfully. A second problem is that, because this attack occurs *before* Election Day, the attacker would not necessarily have the flexibility to adjust her attack to new facts learned immediately before or on Election Day (such as changes in the dynamics of the race, including which candidates are running or how many votes are likely to be needed to ensure a particular outcome). This type of attack is discussed in "Software Attacks on Voting Machines," *infra* at pp. 30–47).

2. Wireless and other Remote Control Attacks. This is also a direct attack on the voting machine. But unlike the "Insertion of Corrupt Software" attack discussed above, this attack can happen on, or immediately before, Election Day (it could also happen much earlier).

This type of attack is often imagined in conjunction with corrupt software attacks. Machines with wireless components are particularly vulnerable to such attacks. Using a wireless PDA or any other device that allows one to access wireless networks, an attacker could instruct a machine to activate (or turn off) a Software Attack Program, send its own malicious instructions, or attempt to read data recorded by the machine.

3. Attacks on Tally Servers. The tally server is a central tabulator which calculates the total votes for a particular jurisdiction (generally at the county level). This attack would occur after the polls have closed and the machines have recorded votes.

An attack on a tally server could be direct (*e.g.,* on the database that totals votes) or indirect (*e.g.,* by intercepting a communication *to* the server). In either case, the attacker would attempt to change or delete the totals reported by the tally server, or the data used to compute those totals.

4. Miscalibration of Machines. All three voting systems use some method to interpret and electronically record the voter's choice. At the close of an election, the machine reports (in electronic and printed form) its tally of the votes. For all three systems, if a machine is not calibrated correctly, it could favor one candidate over another.

We can use the DRE as an example. Let us return to the governor's race in Pennasota: in that race, a touch on the left half of the DRE screen should be recorded as a vote for Tom Jefferson; a vote on the right half of the screen should be recorded as a vote for Johnny Adams. The DRE could be miscalibrated so that touches on the left side, close to the center of the screen, are recorded for Johnny Adams rather than Tom Jefferson.

An obvious problem with this specific example is that most voters who pressed "Jefferson" close to the center of the screen would note on the confirmation screen that their vote had been misrecorded; they would reject the Adams vote and try again. But some might not notice their vote was misrecorded. In these cases, the miscalibration would take votes away from Jefferson and add votes to Adams' total.

5. Shut off Voting Machine Features Intended to Assist Voters. This is another attack that is directed at the machine itself. For all three systems, there are many features that are intended to assist voters ensure that their choices are recorded correctly. By disabling one of these features, an attacker can ensure that some votes would not be accurately recorded.

By way of example, let us return to Pennasota, but this time consider the PCOS machine. PCOS machines have an over/under vote protection that is intended to make sure that voters vote in every race. If a voter accidentally votes for two candidates in the governor's race, the scanner should return the ballot to her without recording any votes. Until she erases one of her choices for governor, or indicates to the machine that she does not want her vote for governor to count, her ballot would not be recorded.

If our attacker is a poll worker who wants Adams to win and works in a polling place where nearly all voters *intend* to vote for Jefferson, she could manually shut off the over/undervote protection. Given the fact that most voters in this polling place want to vote for Jefferson, the chances are that Jefferson would lose some votes as a result. As with the miscalibration attack, this attack does not have to be manual; a Software Attack Program inserted before Election Day could also attempt to shut off such machine functions.

6. Denial of Service Attacks. This covers a broad range of attacks. In essence, this attack is meant to keep people from voting, by making it difficult or impossible to cast a vote on a machine. The attack could be lodged directly upon the machine: for instance, by insertion of corrupt software, as discussed above, or by physically destroying a machine or machines.

Again, looking at the governor's race in Pennasota, our attacker would likely target machines and polling places where she knows most voters would support Tom Jefferson.

7. Actions by corrupt poll workers or others at the polling place to affect votes cast. In our catalogs, these attacks range from activating a Software Attack Program already inserted into a voting machine, to shutting off voting machine functions (discussed above), to giving poor instructions or misleading information to certain voters. It could involve an attack on the machines themselves, upon voters, or upon information meant to be transported from polling places to tally centers. This attack could also include providing incomplete or inaccurate instruction to poll workers.

8. Vote Buying Schemes. This type of attack was already discussed, *infra* at pp. 10–11. As noted, such attacks would require so many informed participants that they are unlikely to affect a statewide election without being exposed.

9. Attacks on Ballots or VVPT. This type of attack could occur at many points. Some jurisdictions purchase their ballots directly from a vendor. Others get their ballots from the county election office. In either case, ballots could be tampered with before they reach the polling place. Both ballots and the VVPT could be tampered with at the polling place, or as they are transported to the county tally center. Finally, in states that have Automatic Routine Audits or recounts of voter-verified paper records, ballots and VVPT could be tampered with prior to the audit at the county offices or tally center.

LESSONS FROM THE CATALOGS: RETAIL ATTACKS SHOULD NOT CHANGE THE OUTCOME OF MOST CLOSE STATEWIDE RACES

The catalogs show us that it is very difficult[60] to successfully change the outcome of a statewide election by implementing “retail” attacks on a large scale. Retail attacks are attacks that occur at individual polling places, or during the transport of hardware and/or ballots to and from individual polling places. We have found that these attacks would require too many participants, and garner too few votes, to have a good chance of swinging a statewide election like the Governor’s race in Pennasota.

In contrast, the least difficult attacks are centralized attacks that occur against the entire voting system. These attacks allow an attacker to target many votes with few fellow conspirators.

The least difficult attacks are centralized attacks that occur against the entire voting system.

To see why retail attacks are unlikely to change the outcome of most close statewide elections, it is useful to look to see how a typical retail threat listed in our catalog might affect the totals in Pennasota’s governor’s race. Attack 20 in the DRE w/ VVPT catalog is the “Paper Trail Boycott” attack.[61] In this attack, an attacker would enlist voters in polling places where her favored candidate is expected to do poorly. Each of the enlisted voters complains to the poll workers that no matter how many times the voter tries, the paper trail record never corresponds to his choices. The election officials would have no choice but to remove the "offending" machines from service. This would reduce the number of available machines, creating a "bottleneck" where voters would have to wait in long lines. Ultimately, some voters would give up and leave the lines without voting.

There is one step to this attack, but it must be repeated many times: voters must falsely complain that the machines are not recording their votes correctly.

Again, we assume that the conspiring voters would want Tom Jefferson to lose a net total of 103,781 votes (there is no switching of votes in this scenario; the attackers hope is that their bottleneck would prevent many of Tom Jefferson's supporters from voting, thus reducing his vote total).

We have assumed that if five voters in a short period of time report that the same machine is not recording their vote correctly, poll workers would be forced to shut it down. As already discussed, the average number of voters per polling place in the State of Pennasota is 1142. Based upon a statistical analysis performed by Professor Benjamin Highton at the University of California at Davis for this report, we estimate that if the attackers shut down three machines in a single polling place, the long lines created by the bottleneck would keep 7.7% of voters from voting in every affected precinct.[62] This means that roughly 88 voters per affected polling place (or 7.7% of 1142) would decide not to vote because of the bottleneck.

But not all of these voters would be Jefferson voters. Even if all of the affected polling places favored Tom Jefferson by 9 to 1, the bottleneck would cause both candidates to lose some votes. Presumably, for every 9 Jefferson voters turned away, 1 Adams voter would also decide not to vote. This means that, if this attack were limited to polling places that heavily favored Tom Jefferson, the effect would be to cause a net loss of 70 votes for Tom Jefferson per polling place (Tom Jefferson would lose 79 – or 90% of the votes in each affected polling place, but Johnny Adams would lose 9 – or 10%).

Based upon this information, we can determine how many polling places would need to be targeted:

> Polling Places Targeted = votes targeted/net number of votes lost by creating bottleneck

or:

> Polling Places Targeted = 103,781/70 = 1,483.

This represents more than one-third of all polling places in Pennasota.[63] It is doubtful that one-third of all polling places in Pennasota would be skewed so heavily toward Jefferson. Professor Henry Brady of the University of California at Berkeley recently performed an analysis of election results in heavily Democratic Broward and Palm Beach counties in the 2000 election. *See*

Appendix I. Even in those counties, only 21.4% and 14.8% of precincts, respectively, reported more than 80% of voters voting for Al Gore; furthermore, only 10.3% and 6.5% (respectively) reported 90% or more voting for Gore.

But even if we were to presume that there were enough polling places to allow this attack to work, there are other problems. First, the attack would probably be exposed: if thousands of machines were reported to have malfunctioned in polling places, but only where Jefferson was heavily favored, someone would probably notice the pattern.

Moreover, the number of informed participants necessary to carry out this attack makes it, in all likelihood, unworkable. The attack would need over 20,000 participants: 5 attackers per machine x 3 machines per polling place x 1,483 polling places.

All other "retail" attacks in the catalog require many hundreds or thousands of co-conspirators. For the reasons already discussed, we believe this makes these attacks very difficult to execute successfully in a statewide election.

In contrast, "wholesale" attacks allow less than a handful of individuals to affect many votes – enough, in some cases, to change the result of our hypothetical governor's race. The least difficult of these wholesale attacks are attacks that use Software Attack Programs. The following section discusses the feasibility of these attacks, which we have identified as the "least difficult" set of attacks against all three voting systems.

A "Trojan horse" is a type of software attack program that "impersonates" a benign program.

SOFTWARE ATTACKS ON VOTING MACHINES[64]

As already discussed, *supra* at p. 6, attacks on elections and voting systems have a long history in the United States. One of the primary conclusions of this report is that, with the new primacy of electronic voting systems, attacks using Trojan horses or other Software Attack Programs provide the least difficult means to affect the outcome of a statewide election using as few informed participants as possible.

This conclusion runs counter to an assertion that many skeptics of these attacks have made, namely that it is not realistic to believe that attackers would be sophisticated enough to create and successfully implement a Software Attack Program that can work without detection. After careful study of this issue, we have concluded that, while operationally difficult, these threats are credible.

HISTORY OF SOFTWARE-BASED ATTACKS

Those skeptical of software attacks on voting machines point to the fact that, up to this point, there is no evidence that a software attack has been successfully carried out against a voting system in the United States. However, the best piece of evidence that such threats should be taken seriously is that, in the last several years, there have been increasingly sophisticated attacks on non-voting computer systems.

Among the targets have been:

- US government systems, including those containing classified data;[65]
- Financial systems, including attacks that gained perpetrators large sums of money;[66]
- Content protection systems intended to stand up to extensive external attack;[67]
- Special-purpose cryptographic devices intended to be resistant to both software and physical attack;[68]
- Cryptographic and security software, designed specifically to resist attack,[69] and
- Attacks on gambling machines, which are subject to strict industry and government regulation.[70]

We learn of more attacks on non-voting systems all the time. But, even with this increased knowledge, we have probably only learned of a small fraction of the attacks that have occurred. For each high-profile case of eavesdropping on cell phones or review of e-mails or pager messages, there are, in all probability, many cases where the attacker's actions remain unknown to the public at large. For every case where financial data is tampered with and the theft is discovered and reported, there are certainly cases where it is never detected, or is detected but never reported.

Many existing attackers have *already shown themselves* to be sophisticated enough to carry out these types of attacks.

In addition to the attacks already listed, we also have seen the rise of sophisticated attacks on widely-used computer systems (desktop PCs) for a variety of criminal purposes that allow criminals to make money:

- Activities/methods like phishing (spam intended to get users to disclose private data that allow an attacker to steal their money) and pharming (exploitation of DNS[71] to redirect legitimate web traffic to illegitimate sites to obtain private data) continue to grow.[72]

- Extortion against some computer sites continues, with an attacker threatening to shut down the site via a distributed denial of services (DDOS) attack, or the posting of confidential information, unless she is paid off.[73]

- Large networks of "bots" – innocent users' computers that have been taken over by an attacker for use in the kinds of attacks already referenced, are bought, sold and rented.[74]

The sophistication of these attacks undermines the argument that attackers "wouldn't be smart enough" to carry out a software attack on voting systems. Many existing attackers have *already shown themselves* to be sophisticated enough to carry out these types of attacks. In fact, given the stakes involved in changing the outcome of a statewide or national election, there is good reason to believe that many who would have an interest in affecting such outcomes are far more sophisticated than recent attackers who have hacked or violated well-protected government and private industry systems.

Still, there are several reasons to be skeptical of software-based attacks, and the rest of this section attempts to address the main challenges an attacker using this method of attack would face:

1. Overcoming Vendor Motivation: The vendor has an economic interest in preventing attackers from infiltrating their machines with Software Attack Programs.

2. Finding an Insertion Opportunity: An attacker would have to gain access to a place that would allow her to insert the Software Attack Program in the machine.

3. Obtaining Technical Knowledge: An attacker would have to know enough to develop a Software Attack Program that can function successfully in a voting terminal.

4. Obtaining Election Knowledge: An attacker may need to know a lot about the ballots and voting patterns of different precincts to create a Software Attack Program that works and does not create undue suspicion.

5. Changing Votes: Once an attacker has sufficient knowledge about the ballots and election, she would need to create a program that can change vote totals or otherwise affect the outcome of an election.

6. Eluding Inspection: An attack would have to avoid detection during inspection.

7. Eluding Testing and Detection Before, During and After the Election: An attacker would have to avoid detection during testing.

8. Avoiding Detection After Polls Close: Even after an attack has successfully changed the electronic record of votes, an attacker would still need to ensure that it is not discovered later.

We review each of these barriers to successful software-based attacks in turn.

VENDOR DESIRE TO PREVENT SOFTWARE ATTACK PROGRAMS

Voting machine vendors have many reasons to want to protect their systems from attack. The most obvious reason is economic: a system that is shown to be vulnerable to attack is less likely to be purchased.

Unfortunately, the fact that vendors have incentives to create secure systems does not mean that their systems are as secure as they should be. The CERT (Computer Emergency Readiness Team) Coordination Center, a federally funded research and development center operated by Carnegie Mellon University, reported nearly 6,000 computer system vulnerabilities in 2005 alone. This included vulnerabilities in two operating systems frequently used on voting machines: 2,328 vulnerabilities on the Linux and Unix operating systems and 812 vulnerabilities in Microsoft Windows operating systems.[75] Many of these vulnerabilities leave machines open to "viruses and other programs that could overtake" them.[76]

Moreover, it is not clear that vendors are doing everything they can to safeguard their systems from attack. As noted in a recent Government Accountability Office report on electronic voting systems, several state election officials, computer security and election experts have criticized vendors for, among other things, their (1) personnel security policies, questioning whether they conduct sufficient background checks on programmers and systems developers; and (2) internal security policies, questioning whether such policies have been implemented and adhered to during software development. [77]

Even assuming that vendors adhere to the strictest personnel and security policies, it is still possible that they would hire employees who abuse their positions to place corrupt software into voting machines. A single, ill-intentioned employee could cause tremendous damage. This is illustrated by the case of Ron Harris, "a mid-level computer technician" for Nevada's Gaming Control Board.[78] Mr. Harris hid a Software Attack Program in dozens of video-poker and slot machines in the early 1990s. The attack program allowed accomplices to trigger jackpots by placing bets in a specific order. Mr. Harris was eventually caught because he became too brazen: by the mid-1990s, he began using an attack program against the gaming machines based on the card game "Keno." When his accomplice attempted to redeem a $100,000 jackpot, officials became suspicious and she was ultimately investigated and caught.[79]

A single, ill-intentioned employee could cause tremendous damage.

In any event, as demonstrated below, an attacker need not be employed at a vendor to insert an attack program into voting machines. She can choose several points to insert her attack, and many of them do not originate at the vendor.

INSERTING THE ATTACK PROGRAM

In this subsection, we look at some of the points where an attacker could insert her attack program. As illustrated by the chart on the next page, the attack program could be inserted while the machine is still in the hands of the vendor, after it has been purchased, and even on Election Day. Insertion into (1) Commercial Off The Shelf (COTS) software used on all voting machines, (2) COTS patches[80] and updates, and (3) ballot definition files,[81] may be particularly attractive because these are not currently subject to inspection by independent testers. Given their size and complexity, it is hard to imagine that a thorough review of them would be practical, even if the COTS vendors were willing to provide access to their source code for inspection.

POINTS OF ATTACK: COTS AND VENDOR SOFTWARE

The process for developing voting system software is not dramatically different from the development of any other type of software or operating systems. Vendors develop a set of requirements for their machines; a team of programmers is subsequently assembled to apply those requirements by developing new code, and then integrating the new code with old code and COTS software; after the new code is written and integrated, a separate team of employees test the machines; when the testers find bugs, they send the new software back to the programmers (which may include new team members) to develop patches for the bugs.

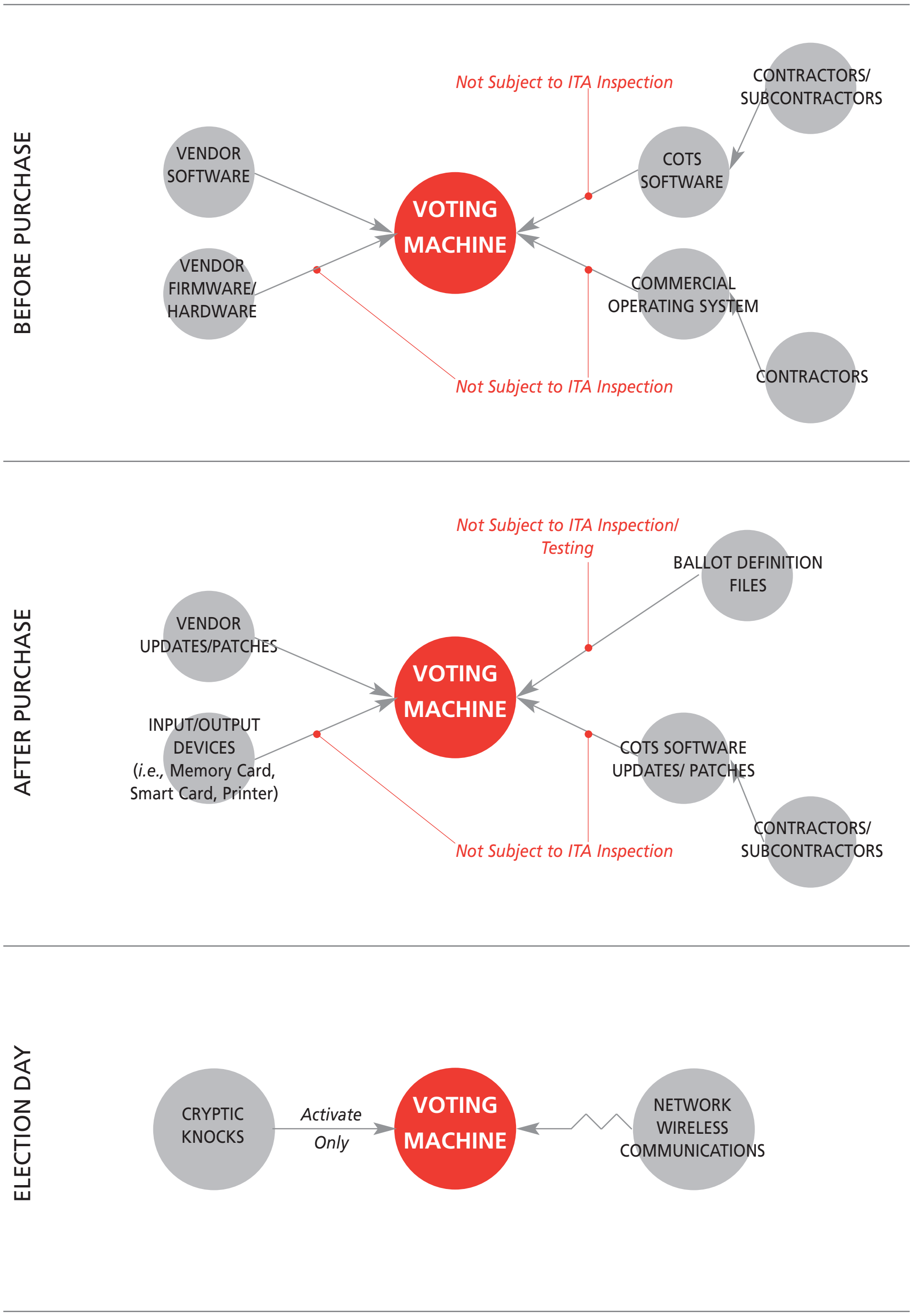
There are a number of opportunities to insert a Software Attack Program during this process:[82]

- The attack program could be part of COTS software that was purchased for use on the voting system. The current voting systems standards exempt unaltered COTS software from inspection by an Independent Testing Authority.[83]

- The attack program could be written into the vendor code by a team member at the vendor.

FIGURE 6

SOFTWARE ATTACK PROGRAM: POINTS OF ENTRY


BEFORE PURCHASE
Not Subject to ITA Inspection
CONTRACTORS/ SUBCONTRACTORS
VENDOR SOFTWARE
COTS SOFTWARE
VOTING MACHINE
VENDOR FIRMWARE/ HARDWARE
COMMERCIAL OPERATING SYSTEM
CONTRACTORS
Not Subject to ITA Inspection
AFTER PURCHASE
Not Subject to ITA Inspection/ Testing
BALLOT DEFINITION FILES
VENDOR UPDATES/PATCHES
VOTING MACHINE
INPUT/OUTPUT DEVICES (i.e., Memory Card, Smart Card, Printer)
COTS SOFTWARE UPDATES/ PATCHES
CONTRACTORS/ SUBCONTRACTORS
Not Subject to ITA Inspection
ELECTION DAY
CRYPTIC KNOCKS
Activate Only
VOTING MACHINE
NETWORK WIRELESS COMMUNICATIONS

- The attack program could be hidden within the operating system using rootkit-like techniques, or perhaps a commercial rootkit for the underlying operating system. [84]

- The attack program could be written into one of the patches that is developed after the vendor's testers find bugs.

- The attack program could be written by someone at the vendor after it has passed the vendor's testing.

It is worth noting that even tampering with the software in the *initial voting system is not limited to programmers working for the voting system vendor.* COTS software writers, who may themselves be contractors or subcontractors of the original company that sold the COTS software to voting systems vendors, are in a very good position to insert an attack program.

Further, anyone with access to the voting system software before it has been installed on the voting machines may install an attack program. This could include people with access to the software during development, storage, or testing.

Anyone with access to the voting system software before it has been installed on the voting machines may install an attack program.

POINTS OF ATTACK: SOFTWARE PATCHES AND UPDATES

COTS software is often supplemented by patches and updates that can add features, extend the software's capabilities (*e.g.,* by supporting more assistive technology or a larger set of screen characters for alternate-language voting) or fix problems discovered after the software was sold. This is an obvious attack point. The attack program may be inserted by someone working for the COTS software vendor, or by someone working at the voting system vendor, or by the election official handling the installation of patches and updates. The patch or update can be installed before or after the voting machine has left the vendor.

POINTS OF ATTACK: CONFIGURATION FILES AND ELECTION DEFINITIONS

As discussed, *infra* at endnote 81, ballot definition files allow the machine to (1) display the races and candidates in a given election, and (2) record the votes cast. Ballot definition files cannot be created until shortly before an election, when all of the relevant candidates and races for a particular jurisdiction are known. An attacker could take over the machine by inserting improperly formed files at the time of Ballot Definition Configuration. Two separate reports have demonstrated that it may be possible to alter the ballot definition files on certain DREs so that the votes shown for one candidate are recorded and counted for another.[85] The Task Force knows of no reason why PCOS systems would not be similarly vulnerable to such an attack.

Ballot definition files are not subject to testing by Independent Testing Authorities and cannot be, because they are developed for specific jurisdictions and elections, after certification of a voting system is complete.[86]

Two separate reports have demonstrated that it may be possible to alter the ballot definition files on certain DREs so that the votes shown for one candidate are recorded and counted for another. The Task Force knows of no reason why PCOS systems would not be similarly vulnerable to such an attack.

POINTS OF ATTACK: NETWORK COMMUNICATION

As will be discussed in greater detail, *infra* at pp. 85–86, some voting systems use wireless or wired network connections. If there is a vulnerability in the configuration of the voting machine (again, by design or error), this can allow an attacker to insert an attack program via the wireless connection.

POINTS OF ATTACK: DEVICE INPUT/OUTPUT[87]

Some voting systems involve the use of an external device such as a memory card, printer, or smart card. In some cases, the ability to use these devices to change votes has been demonstrated in the laboratory. For example, Harri Hursti, a member of the Task Force, has demonstrated that memory cards (which generally contain, among other things, the ballot definition files) can be used to create false vote totals on a particular brand of PCOS, and conceal this manipulation in reports generated by the scanners to election officials.[88] This was recently demonstrated again in a test performed by election officials in Leon County, Florida.[89] Several computer security experts that have reviewed other PCOS systems believe that they may be vulnerable to similar attacks.[90]

DREs have also been shown to be vulnerable to attacks from input devices. In a "Red Team" exercise[91] for the State of Maryland in January 2004, RABA Technologies, LLC, demonstrated that smart cards (which are used as both supervisor and voter access cards) on one model of DRE could be manipulated to allow a voter to vote multiple times.

TECHNICAL KNOWLEDGE

Just because there are opportunities to insert a Software Attack Program does not mean that an attacker would have the knowledge to create a program that works. It is not difficult to understand how hackers could gain enough knowledge to create attack programs that could infiltrate common operating systems on personal computers: the operating systems and personal computers are publicly available commercial products. A hacker could buy these products and spend months or years learning about them before creating an effective attack program.

How would an attacker gain enough knowledge about voting systems to create an attack program that worked? These are not systems that general members of the public can buy.

We believe there are a number of ways an attacker could gain this knowledge. First, she might have worked for (or received assistance from someone who worked for) one of the voting system vendors. Similarly, she could have worked for one of the independent testing authorities or state qualification examiners.

Alternatively, the attacker could hack into vendor or testing authority networks. This could allow her to gain important knowledge about a voting machine's software and specifications.

Finally, an attacker could steal or "borrow" a voting machine. Access to voting machines will be very important to an attacker as she develops her Software Attack Program; this will not necessarily be an overwhelming obstacle. Machines are often left in warehouses and polling places for months in between elections. Responses to our security surveys showed that there are many points where physical security for voting machines is surprisingly lax: about half of the counties responding to the security survey stated that they did not place tamper evident seals on machines during the months the machines were in storage; several counties stated that they did not take inventory of voting machines in between elections; in one county, voting machines were placed under a blanket in the back of an office cubicle when not in use.[92] Hackers have repeatedly shown their ability to decipher software and develop attack programs by "reverse engineering" their target machines; there is no reason to believe they could not apply these skills to voting machines.[93]

Responses to our security surveys showed that there are many points where physical security for voting machines is surprisingly lax.

ELECTION KNOWLEDGE

An attacker could be required to insert the Software Attack Program before all facts about the election are known. Many points of insertion discussed above (*supra* at pp. 33–35) would require the attacker to create an attack program before she could possibly know which candidates were running or where various races would be placed on ballots. Different jurisdictions could decide to place that same race in different positions on the ballot (*i.e.,* as the third race as opposed to the fourth).

ATTACKING THE TOP OF THE TICKET

We believe this problem could be overcome, particularly where the attacker sought to shift votes at the "top" of the ticket – as would be the case in an attempt to affect the governor's race in Pennasota in 2007. Here, in a software update or patch that is sent before a particular election, the attacker could merely ask the machine to switch one or two votes in the first race in the next election. Since the Federalists and the Democratic-Republicans are the two main parties in Pennasota, the attacker would know that their candidates for governor would be listed in the first and second columns in the governor's race. Even if the attacker is not certain whom the Federalists or Democratic-Republicans are going to select as candidates at the time when she inserts the attack program, she could still create a successful program by instructing the machine to switch a certain number of votes in the first (governor's) race from the Democratic-Republicans (column "2") to the Federalists (column "1").

Moreover, we have assumed that our attacker is smart enough to avoid switching so many votes that her attack would arouse suspicion. By switching 7.5% or fewer votes per machine, our attacker need not be particular about which machine she attacks. She could create a program that only activates on every fourth or fifth machine.

PARAMETERIZATION

It is possible that our attacker would be more cautious: perhaps she would limit her attack to certain counties or precincts. Perhaps in some jurisdictions the governor's race won't be listed as the first race. Or perhaps her opportunity to insert the attack program came a year before the governor's race, when she wasn't sure who the candidates would be and whether she would want to attack the election.

In such cases, the attacker could "parameterize" her attack. Under this scenario, the attacker would create an attack program and insert it in the original software, or software updates. The attack program would not specify which race to attack or how. Instead, it would wait for certain commands later; these commands would tell it which votes to switch.

These commands could come from many sources, and could be difficult for anyone other than the attacker to find. For instance, the commands could come from the ballot definition file.[94] The original attack program could provide that if there is an extra space after the last name of the second candidate for a particular race in a ballot definition file, five votes in that race should be switched from the second column to the first. By waiting to provide these commands until the ballot definition files are created, the attackers could affect a race with great specificity – instructing the attack program to hit specific precincts in specific ways.

Of course, this is a more difficult attack: it requires more steps and more informed participants (both the original programmer and the person to insert the commands in the ballot definition file). In the specific example we have provided, it would also require someone with insider access to the ballot definition files.

But this type of attack would be attractive because it would give the attacker a great deal of flexibility. Moreover, the commands could come from sources other than the ballot definition files. If the voting machines have wireless components, the attacker could activate her attack by sending commands over a wireless PDA[95] or laptop. Or she could send these commands through a Cryptic Knock[96] during, for instance, voting or Logic and Accuracy testing.[97] For example, an insider responsible for developing the Logic and Accuracy scripts could have all the testers type in a write-in candidate for the ostensible purpose of ensuring that the write-in function is working. The spelling of the name of that write-in candidate could encode information about what races and ballot items should be the target of the attack. Testers following the script would unknowingly aid the attack.

CREATING AN ATTACK PROGRAM THAT CHANGES VOTES

Even if the attacker possessed sufficient knowledge about voting systems and specific elections before she inserted her attack program, she would need to figure out a way to create a tampering program that alters votes.[98] Without getting into the fine details, this subsection will summarize a number of methods to accomplish this goal.

CHANGING SYSTEM SETTINGS OR CONFIGURATION FILES

Configuration Files are files that are created to organize and arrange the system settings for voting machines. The system settings control the operation of the voting machine: for instance, setting parameters for what kind of mark should count as a vote on the PCOS ballot, instructing the PCOS scanner to reject ballots that contain overvotes, setting parameters for dividing a DRE screen when there are multiple candidates in the same race, or providing a time limit for voters to cast their votes on DREs.

An attack program that altered the system settings or Configuration Files could be buried in a Driver or program that is only run when the voting has started, or work off of the voting machine clock, to ensure that it is triggered at a certain time on Election Day. Among the attacker's many options within this class of attack are:

- Swap contestants in the ballot definition or other files, so that, for instance, a vote for Tom Jefferson is counted as one for Johnny Adams (and vice versa). This is an attack that was described in the RABA Technologies report on an intrusion performed for the state of Maryland.[99]

- Alter Configuration Files or system settings for the touch-screen or other user interface device, to cause the machine to cause differential error rates for one side. For instance, if our attacker knew that voters for Tom Jefferson were more likely to overvote or undervote the first time they filled out their ballots, she could install a software attack that shut off the overvote/undervote protection in several PCOS scanners – *see infra* p. 81 for a discussion of this attack.

- Alter Configuration Files or system settings to make it easier to skip a contest or misrecord a vote accidentally (*e.g.*, by increasing or decreasing touch-screen sensitivity or misaligning the touch-screen).

- Alter Configuration Files or system settings to change the behavior of the voting machine in special cases, such as when voters flee (for instance, recording a vote for Johnny Adams when a voter leaves the booth without instructing the machine to accept her ballot).

The attack that introduces biased errors into the voter's interaction with the voting system is especially useful for attacking DRE w/ VVPT and PCOS systems since the attacked behavior, if detected, is indistinguishable from user error.

There are at least two potential operational difficulties an attacker would have to overcome once she inserts this type of attack program: (1) she would need to control the trigger time of the attack so as to avoid detection during testing; and (2) she would want to make sure that the changes made are not entered into the Event Logs, in case they are checked after the polls have closed. Ways of overcoming these challenges are discussed *infra* at pp. 41–44 and 45–46.

ACTIVE TAMPERING WITH USER INTERACTION OR RECORDING OF VOTES

In this type of attack, the attack program triggers during voting and interferes in the interaction between the voter and the voting system. For example, the attack program may:

- Tamper with the voter interaction to introduce an occasional "error" in favor of one contestant (and hope that the voter does not notice). This is the "Biased Error" attack.

- Tamper with the voter interaction both at the time the voter enters his vote and on the verification screen, so that the voter sees consistent feedback that indicates his vote was cast correctly, but the rest of the voting machines software sees the changed vote.

- Tamper with the electronic record written after the verification screen is accepted by the voter – *e.g.*, by intercepting and altering the message containing results before they are written in the machine's electronic record, or any time before end-of-election-day tapes (which contain the printed vote totals) are produced and data are provided to election officials.

This class of attack seems to raise few operational difficulties once the attack program is in place. The attack that introduces biased errors into the voter's interaction with the voting system is especially useful for attacking DRE w/ VVPT and PCOS systems where the paper record is printed or filled in by the voting machines being attacked, since the attacked behavior, if detected, is indistinguishable from user error. However, the attack program could improve its rate of successfully changed votes, and minimize its chances of detection, by choosing voters who are unlikely to check their paper records carefully. Thus, voters using assistive technology are likely targets.

TAMPERING WITH ELECTRONIC MEMORY AFTER THE FACT

An alternative approach is to change votes in electronic memory after voting has ended for the day, but before the totals are displayed locally or sent to the county tally server.

In this case, the attack program need only be activated after voting is complete.

This allows the attack program considerable flexibility, as it can decide whether to tamper with votes at all, based on totals in the machine. For instance, the Software Attack Program could be programmed to switch ten votes from Tom Jefferson to Johnny Adams, only if Johnny Adams has more than 90 votes on the machine.

It can also allow the attack program to avoid getting caught during pre-election testing. By programming the attack program to activate only after voting has ceased on Election Day (and the program should be able to do this by accessing the voting machine's internal clock), the attack program would elude all attempts to catch it through earlier testing. Similarly, by only triggering after, for instance, 100 votes have been cast within twelve hours, the attack program can probably elude pre-election testing; most pre-election testing involves the casting of far fewer votes. *See infra* p. 107.

This type of attack must overcome some interesting operational difficulties; we do not believe that any of them are insurmountable with respect to any of the systems we have reviewed:

- Some voting machines store electronic records in several locations; the attack program would have to change them all.

- The attack program must either (1) avoid leaving entries of attack in the event or Audit Logs, or (2) create its own Audit Logs after the attack (however, the necessity of doing either of these things is dependent upon how the machine logs its own actions: if the machine would show only that it accessed a file, these are unlikely to be problems for the attack program; if each record altered yields a log entry, this requires tampering with the event log to avoid detection).

- Depending upon details of the file access required, the attack program may face some time constraints in making the desired number of changes. Given the fact that we have assumed no more than 7.5% of votes would be switched in any one polling place or 15% on any machine, this may not be a great problem. There is likely to be a reasonable span of time between the closing of polls and the display and transmission of results.

Attacks installed at certain points may not be subject to any inspection.

ELUDING INDEPENDENT TESTING AUTHORITY INSPECTIONS[100]

How does an attacker ensure that an attack program she has inserted would not be caught by inspections[101] done at the vendor, or during an Independent Testing Authority inspection of software code?

Part of the answer depends upon where the attack program is installed. Attacks installed at certain points (such as attacks written into vendor software code) are likely to be subject to multiple inspections; attacks installed at other points (such as attacks installed in COTS software, ballot definition files or replaceable media) may not be subject to any inspection.

CREATE DIFFERENT HUMAN-READABLE AND BINARY CODE[102]

A clever attacker could defeat inspection in a number of ways. Before detailing how this would be accomplished, a brief conceptual introduction is necessary: To develop a program, a programmer writes human-readable source code. Generally, before a computer can run this program, the source code must be converted into a binary code (made up of "0"s and "1"s) that the computer can read. This conversion is accomplished by use of a compiler.[103] Thus, each program has two forms: the human-readable source code and the compiled binary code.

A simple attack designed to elude inspection could be accomplished as follows: our attacker writes human-readable source code that contains an attack program (perhaps the program, among other things, instructs the machine to switch every 25th vote for the Democratic-Republicans to the Federalists). The attacker then uses a compiler to create a similarly malicious binary code to be read by the computer. After the malicious binary code has been created, the attacker replaces the malicious human-readable source code with a harmless version. When the vendor and Independent Testing Authority inspect the human-readable source code, they would not be able to detect the attack (and the binary code would be meaningless to any human inspector).

USE ATTACK COMPILER, LINKER, LOADER OR FIRMWARE

An obvious way for an ITA to pre-empt this attack would be to require vendors to provide the human-readable source code, and to run the human-readable source code through the ITA's compiler. The ITA could then compare its compiled version of the code with the compiled code provided by the vendor (*i.e.*, did all the "0"s and "1"s in both versions of the code match up?).

But what if, instead of inserting the attack into the vendor's source code, our attacker inserted an attack into the compiler (which is generally a standard software program created by a non-voting system software vendor)? Under these circumstances, the compiler could take harmless human-readable source code and turn it into malicious binary code without any inspector being the wiser. As a compiler is generally COTS software, it would not be inspected by the ITAs.

In any event, the attacker could hide the attack program in the compiler by adding one level of complexity to her attack: make the compiler misread not only the seemingly innocuous vendor source code (which would be converted into malicious binary code), but also the seemingly innocuous compiler source code (which would also be converted into malicious binary code, for the purpose of misreading the vendor source code). In other words, the attacker can hide the attack program in the same way that she might hide an attack program in other software: change the human-readable compiler source code so that it does not reveal the attack. When the compiler "compiles itself" (*i.e.,* turning the human-readable source code for the compiler into computer readable binary code) it creates a binary code that is malicious, but cannot be detected by human inspectors.

The compiler is not our attacker's only opportunity to convert innocuous human-readable source code into an attack program. What is known as a "linker" links the various binary code programs together so that the voting machine can function as a single system. Here again, the linker can be used to modify the binary code so that it functions as an attack program.

Additionally, the attacker can use the "loader" – the program on each voting machine's operating system that loads software from the disk drive onto the machine's main memory, to alter code for a malicious purpose.[104]

Finally, if our attacker is a programmer employed at the vendor, she can create or alter firmware[105] that is embedded in the voting machines' motherboard, disk drives, video card or other device controllers to alter seemingly harmless code to create a malicious program. Like COTS software, firmware is not subject to ITA inspection.

AVOIDING INSPECTION ALTOGETHER

An attacker could also insert her program in places not subject to inspection.

As already noted, the current Voluntary Voting Systems Guidelines exempts unaltered COTS software from testing, and original COTS code is not currently inspected by the ITAs.[106] This makes it more difficult to catch subtle bugs in either COTS software that is part of the original voting system, or COTS software patches and updates (assuming that new testing is done when such patches and updates are required).

Moreover, attacks inserted through ballot definition, via wireless communication, or through device input (*i.e.,* memory cards, printers, audibility files) would occur after the machine has been tested by the ITA and would thus avoid such testing altogether.

Moreover, we have serious concerns about the ability of current Independent Testing Authority inspections and tests to catch even Software Attack Programs and bugs in original voting systems software. While ITA tests may filter out obvious attack behavior, intentional, subtle bugs or subtle attack behavior (*e.g.*, triggering the attack behavior only after complicated interaction with a user unlikely to be replicated in a testing lab, or only when the clock tells the Attack Program that it is Election Day) may remain unnoticed in the testing lab review. As noted in the GAO report, these and other concerns about relying on ITA testing have been echoed by many security and testing experts, including ITA officials.[107]

AVOIDING DETECTION DURING TESTING

Even after an attack program has been successfully installed and passed inspection, it would still need to get through testing. Tampered software must avoid detection during testing by vendors, testing authorities and election officials. With the exception of Parallel Testing (which is regularly performed statewide only in California, Maryland, Washington), all of this testing is done prior to voting on Election Day.[108]

There are a number of techniques that could be used to ensure that testing does not detect the attack program.

- The attack program could note the time and date on the voting machine's clock, and only trigger when the time and date are consistent with an election. This method could, by itself, prevent detection during vendor testing, Logic and Accuracy Testing and Acceptance Testing, but not during Parallel Testing.

- The attack program could observe behavior that is consistent with a test (as opposed to actual voter behavior). For example, if Logic and Accuracy Testing is known never to take more than four hours, the attack program could wait until the seventh hour to trigger. (Note that the attack becomes more difficult if the protocol for testing varies from election to election).

- The attack program could activate only when it receives some communication from the attacker or her confederates. For example, some specific pattern of interaction between the voter or election official and the voting machine may be used to trigger the attack behavior. This is often called a "Cryptic Knock."

AVOIDING DETECTION AFTER THE POLLS HAVE CLOSED

In many cases, the most effective way to tamper with an election without detection would be to change votes that have actually been cast; this way, there would be no unusual discrepancy between the poll books (which record the number of voters who sign in) and vote totals reported by the machines.[109] In the case of a DRE system, changing votes electronically changes all official records of the voter's choice, so this kind of attack cannot be directly detected by comparing the electronic totals with other records. In the case of other voting systems, such as DRE w/ VVPT or PCOS, the attacker must also tamper with the paper records, or prevent their being cross-checked against the electronic records, *assuming that there is some policy in place that requires jurisdictions to check paper records against the electronic totals.*

In the case of a DRE system, changing votes electronically changes all official records of the voter's choice, so this kind of attack cannot be directly detected by comparing the electronic totals with other records.

DECIDING HOW MANY VOTES TO CHANGE

An attack could be detected if there were a very strong discrepancy between informal numbers (polling data, or official results in comparable precincts or counties) and reported election results. There are at least a couple of ways that an attack program could minimize suspicion from this kind of evidence:

- Where possible, the attack program on the voting machines would change a fixed portion of the votes (for instance, in the attack scenarios we have developed, we have assumed that no more than 7.5% of votes in any single polling place would be switched), rather than simply reporting a pre-ordained result. This avoids the situation where, for instance, a recently indicted candidate mysteriously wins a few precincts by large margins, while losing badly in all others, raising suspicion that there was an attack. It also prevents a situation where a candidate wins 80-90% of the vote in one polling place, while losing badly in all other demographically similar polling places.

- The attack program might also detect when the tampering is hopeless (*e.g.*, when the election appears so one-sided that the benefit of improving the favored candidate's outcome is outweighed by the cost of increased chance of detection from implausible results). In that case, it would refrain from any tampering at all, since this would risk detection without any corresponding chance of success.

AVOIDING EVENT AND AUDIT LOGS

Tampered software must not leave telltale signs of the attack in any Event or Audit Logs.[110] There are a number of ways the attack program could accomplish this goal, depending upon the nature of the attack program and the software it targets:

- Tampered user-interface software could display the wrong information to the voter (meaning the voter believes his vote has been recorded accurately), while recording the attack program choice in all other system events. In this case, there would be no trace of the attack in the event log.[111]

- Tampered Driver software for storage devices or tampered BIOS[112] could alter what is written to the storage devices.

- A tampered operating system or other high-privilege-level software could tamper with the logs after entries are made, avoiding record of such an attack in the logs.[113]

- A tampered operating system or other software could provide a different log to the outside world than the one stored internally, if the log is not stored on removable media.

COORDINATING WITH PAPER RECORD ATTACKS[114]

When the attacker must also tamper with paper records (*i.e.*, in the case of PCOS and DRE w/ VVPT systems), she would likely need to prepare replacement paper records before the voting is completed.[115]

This coordination task could be solved in a number of ways:

- The attacker could wait until the election is over, and then print the replacement paper records. This raises some logistical problems for the attacker, such as how to find out what the electronic records show, and print enough paper records once this information is learned and replace the paper.

- If the attacker is in contact with the voting machine during the voting process – for example over a wireless network or via an exposed infrared port – the attacker could print replacement paper records as the tampered records are produced on the voting machine.

- The attack program could have a predefined sequence of votes, which it produces electronically and which the attacker can print at any time.

- The attacker could communicate with the voting machine after voting has ended but before the votes have been displayed to poll workers or sent to the tabulation center. In this case, the attacker could tell the voting machine what totals to report and store. This could be done remotely (via wireless or exposed infrared port) or through some form of direct interaction with the machine (this would obviously require many conspirators if multiple machines were involved).

In all cases, the attacker would have the additional problem of replacing the original records with her created paper records. We discuss this issue *infra* at pp. 71–75.[116]

CONCLUSIONS

Planting a Trojan Horse or other Software Attack Program, though operationally challenging, is something that a sophisticated attacker could do. An attacker could take advantage of several points of vulnerability to insert corrupt software. Many of these points of vulnerability are currently outside the testing and inspection regimen for voting systems. In any event, we are not confident that testing and inspection would find corrupt software even when that software is directly tested and inspected by an ITA.

Our attacker – who aims to move roughly 52,000 votes from the Democratic-Republicans to the Federalists in the gubernatorial race in Pennasota – need not know much about the particulars of the election or about local ballots to create an effective attack program, and thus could create her attack program at almost any time. To the extent she is concerned about the names of the candidates or particulars of local ballots, however, she could parameterize her attack by, for instance, inserting instructions into the ballot definition files or sending instructions over a wireless component, when she would have all the information she could want about local ballots.

There are a number of steps – such as inspecting machines to make sure that all wireless capabilities are disabled – that jurisdictions can take to make software attacks more difficult. Ultimately, however, this is a type of attack that should be taken seriously.

A software attack allows a single knowledgeable person (or, in some cases, small group of people) to reach hundreds or thousands of machines.

LEAST DIFFICULT ATTACKS APPLIED AGAINST EACH SYSTEM

As already discussed, in a close statewide election like the Pennasota governor's election, "retail" attacks, or attacks on individual polling places, would not likely affect enough votes to change the outcome. By contrast, the less difficult attacks are centralized attacks: these would occur against the entire voting system and allow an attacker to target many votes with few informed participants.

Least difficult among these less difficult attacks would be attacks that use Software Attack Programs. The reason is relatively straightforward: a software attack allows a single knowledgeable person (or, in some cases, small group of people) to reach hundreds or thousands of machines. For instance, software updates and patches are often sent to jurisdictions throughout a state.[117] Similarly, replaceable media such as memory cards and ballot definition files are generally programmed at the county level (or at the vendor) and sent to every polling place in the county.

These attacks have other benefits: unlike retail denial of service attacks, or manual shut off of machine functions, they could provide an attacker's favored candidate with a relatively certain benefit (*i.e.*, addition of "x" number of votes per machine attacked). And if installed in a clever way, these attacks have a good chance of eluding the standard inspection and testing regimens currently in place.

Below, we look at examples of these least difficult attacks against each system: how they would work, how many informed participants would be needed, how they might avoid detection, and how they could swing a statewide election. In addition, we evaluate the effectiveness of each of the three sets of countermeasures against them.

ATTACKS AGAINST DRES WITHOUT VVPT

The Task Force has identified over thirty-five (35) potential attacks against DREs without VVPT.[118] All of the least difficult attacks against DREs without VVPT involve inserting Software Attack Programs into the DREs. In this section, we will examine an example of this least difficult attack and how much more "expensive" such attacks are made by the "Basic Set" and "PT Set" of countermeasures. *We cannot examine the "ARA Set" of countermeasures against these attacks, because DREs do not have a voter-verified paper trail to allow auditing to occur.*

We are also particularly concerned about attacks that are made easier by use of wireless networks. This set of attacks will be examined here under "Other Attacks of Concern."

REPRESENTATIVE "LEAST DIFFICULT" ATTACK: TROJAN HORSE INSERTED INTO OPERATING SYSTEM (DRE ATTACK NO. 4)

As already discussed, there are several potential points of entry for a Software Attack Program. We could have chosen any number of Software Attack Programs in our DRE Attack Catalog. We have chosen Attack Number 4, "Trojan Horse Inserted into Operating System," because it is representative of these attacks and easy to explain.

As already discussed, a "Trojan Horse" is a type of Software Attack Program that masquerades as a benign program component. Unlike viruses, Trojan Horses do not replicate themselves.

DESCRIPTION OF POTENTIAL ATTACK

Here is how this representative attack works:[119]

- A third-party software company supplies a publicly available operating system for DREs.[120]

- As already noted, the Trojan Horse could be inserted by any number of people: a programmer working for the voting system vendor, the operating system vendor, or an employee of a company that contracts with the software company that creates the operating software.[121] The Trojan Horse could also be inserted in an operating system update or patch that would be inserted on any voting machine that ran on this operating system.[122]

- The attacker could change the human-readable source code for the operating system, to ensure that anyone who decided to inspect the code would not find the Trojan Horse. In any event, the operating system is COTS software, so it is unlikely to be reviewed by the vendor, or inspected by the ITA.

- The Trojan Horse is coordinated with the voting machine's internal clock and set to activate after ITA, Acceptance and Logic and Accuracy Testing are complete (*e.g.*, the first Tuesday after the first Monday in November 2007, after 11 a.m.). This would prevent any detection during such testing.

- Among the many ways a Trojan Horse could ensure the misrecording of votes, it could

  - detect when a ballot is displayed, and reverse the order of the first two entries on the screen (so if the order should be, for example, John Adams and Tom Jefferson, the displayed order is Tom Jefferson and John Adams). In this scenario, the Trojan Horse would also check for the names on the review screen, and if either of the two names appeared, the other would be substituted and recorded.

- alter votes in the electronic memory at the end of a full day of voting. This might be slightly more complicated, as it could require the Trojan Horse to change the electronic records in the many locations where vote totals are stored, and avoid leaving entries in the Event and Audit Logs, or create new logs.
- display information as the DRE is intended to (*i.e.*, ballot positions are not reversed and verification screens let voters believe their choices have been accurately recorded), but record the Trojan Horse's choice in all other system events.

- The Trojan Horse can attempt to ensure that no one would discover what it has done after the election is over, even if there are suspicions that machines were attacked.
  - It could tamper with the Event and Audit logs after the attack is complete, preventing the creation of a record of such an attack in the logs.
  - It could create and provide a new log to the outside world, different than that stored internally.
  - It could avoid the Event and Audit Logs altogether, by displaying the wrong information to the voter (*i.e.*, allowing the voter to believe his voter has been recorded correctly), while recording the Attack Program's choice in all other system events.

We estimate that with clever enough attackers, this attack could successfully be completed with just one person; this attack involves only one step: design and insertion of the Trojan Horse.[123] Obviously, it would be important for the designer of the Trojan Horse to understand the workings of the DRE she seeks to attack.[124] But once the Trojan Horse was successfully inserted, it would not require any further involvement or informed participants.

HOW THE ATTACK COULD SWING STATEWIDE ELECTION

In the race for governor of Pennasota, 3,459,379 votes would be cast, and the election would be decided by 80,257 votes (or 2.32%). We assume that the attacker would want to leave herself some margin of error, and therefore aim to (1) add 103,781 votes (or 3%) to Johnny Adams's total (or subtract the same from Tom Jefferson) or (2) switch 51,891 votes from Tom Jefferson to Johnny Adams.

As we assume that each DRE would record roughly 125 votes, we calculate that Pennasota would have approximately 27,675 DREs.[125] This would require the

Software Attack Program *to switch fewer than 2 votes* per machine to change the outcome of this election and do so with a comfortable margin of victory.[126]

EFFECT OF BASIC SET OF COUNTERMEASURES

The Basic Set of Countermeasures that apply to DREs without VVPT are as follows:

- the model of DRE used in Pennasota has passed all relevant ITA inspections;
- before and after Election Day, machines for each county are locked in a single room;
- some form of "tamper evident" seals are placed on machines before and after each election;
- the machines are transported to polling locations during the five to fifteen days before Election Day;
- acceptance testing is performed by every county at the time the machines are delivered from the vendor;
- Logic and Accuracy Testing is performed immediately prior to each election by the County Clerk;
- at the end of Election Day, vote tallies for each machine are totaled and compared with the number of persons who have signed the poll books; and
- a copy of totals for each machine is posted at each polling place on election night and taken home by poll workers to check against what is posted publicly at election headquarters, on the web, in the papers, or elsewhere.

Given the small number of votes changed per machine, we do not believe that the altered machine totals alone would alert election officials or the public to the fact that election results had been changed.

As already explained, *supra*, at pp. 42–43, there is a good chance that the ITA (and, for that matter, the vendor) would not find the attack during its inspection of the code. First, the attacker could erase the Trojan Horse from the human-readable source code, on the chance that an inspector might review the operating system's source code carefully. In this case, only a careful forensic analysis of the machine could find the Trojan Horse. Second, because the operating system is COTS code, it is unlikely that the code for the operating system (and its updates and patches) would be inspected at all.[127] Third, if the Trojan Horse is part of an operating system update or patch, it may never even enter an ITA. The model would have already passed inspection; it is unlikely that local jurisdictions or the vendor would ask the ITA to conduct an entirely new test and inspection with a model that has the COTS patch or update installed.

Once the Trojan Horse was inserted, the physical security detailed in the Basic Set of Countermeasures would not be of any benefit.

Finally, the testing done in this set of countermeasures should not catch the attack. The Trojan Horse, by waiting until 11 a.m. on Election Day, would ensure that all testing is complete. Posting election night results at the polling place would not help either; these results would match county election totals. Unfortunately, neither set of numbers would match actual voter choice.

Based on this analysis, we have concluded that the Basic Set of Countermeasures would not require our attacker to add any more informed participants to complete her attack successfully.

EFFECT OF REGIMEN FOR PARALLEL TESTING

As already discussed, the Regimen for Parallel Testing involves selecting voting machines at random and testing them as realistically as possible during the period that votes are being cast. The object of this testing is to find any bug (whether deliberately or accidentally installed) that might be buried in the voting machine software, and which could affect the ability of the voting machines to record votes accurately. Unlike other pre-election testing which is almost always done using a special "test mode" in the voting system, and thus might be subverted by a clever attacker relatively easily, Parallel Testing attempts to give no clues to the machine that it is being tested. Professional testers cast votes generated by a script for the full Election Day (this would allow the testers to find an attack that triggers, for example, after 11 a.m. on Election Day). If Parallel Testing is done as we suggest, these cast votes would simultaneously be recorded by a video camera. At the end of the day, election officials reconcile the votes cast on the tested machine with the results recorded by the machine. The video camera is a crucial element in the Regimen for Parallel Testing, because it allows officials to ensure that a contradiction between the machine record and the script is not the result of tester error.

The Trojan Horse attack is one of the attacks that Parallel Testing is intended to catch.[128] There should be no question that if properly implemented, Parallel Testing would make a Trojan Horse attack more difficult.

But how much more difficult, and in what way? In the following subsections, we assess the ways an attacker might subvert Parallel Testing and how difficult this subversion would be: this includes a review of the ways in which Parallel Testing may force an attacker to invest more time, money and technical savvy to implement a least difficult attack like DRE Attack Number 4 successfully. It also includes an assessment of the number of additional informed participants that would be needed to implement this attack when the Regimen for Parallel Testing

Plus Basic Set of Countermeasures is in place.

We have identified two ways that an attacker might be able to subvert Parallel Testing, and thus still successfully implement DRE Attack Number 4. They are:

1. infiltrate the Parallel Testing teams; and

2. create an Attack Program that can recognize when it is being Parallel Tested, and knows to shut off under such circumstances.

As discussed in further detail below, in certain scenarios, an attacker could combine these two methods to subvert Parallel Testing.

A state does not have to test a particularly large number of machines for Parallel Testing to catch a Trojan Horse that has been inserted for the purpose of changing the outcome of a statewide election.

Infiltrating the Parallel Testing Teams

Subverting Parallel Testing by simply infiltrating the Parallel Testing team would be extremely difficult. To have a reasonable chance of defeating Parallel Testing this way, the attacker would have to add approximately 100 informed participants to her conspiracy.[129]

As detailed in Appendix I¸ a state does not have to test a particularly large number of machines for Parallel Testing to catch a Trojan Horse that has been inserted for the purpose of changing the outcome of a statewide election. Assuming that 5% of all machines were attacked (and in Pennasota, we have determined that more than 7% would need to be attacked to swing a statewide election), [130] we calculate that it would be necessary to test only 58 machines to have a 95% chance of testing a machine that has a Trojan Horse.[131] As a reminder, under the Regimen for Parallel Testing Plus Basic Set of Countermeasures, every testing team would have a tester and observer.

If the attacker managed to infiltrate half of the teams of testers and observers and make them part of the attack conspiracy (for a total of 58 additional fully informed participants), there would still be a greater than 75% chance that uncorrupted parallel testers would test a machine that contained a Trojan Horse.[132]

To reduce the odds substantially, so that there would be just a 10% chance that Parallel Testing would catch the Trojan Horse, the attacker would need to involve all but two teams of testers (meaning she would need to involve over 100 parallel testers and observers in the conspiracy).[133]

Creating an Attack That Recognizes Testing

Rather than subvert Parallel Testing by infiltrating the Parallel Testing teams alone, an attacker's best bet would be to design an attack program that is capable of recognizing when it is being tested. How might a tampered program detect that a machine is being used by parallel testers, rather than by regular voters? There are a number of possibilities:

a. *Warning the Trojan Horse:* Someone on the attack team may communicate with the Trojan Horse, for example by using a wireless data channel to deactivate the attack behavior when Parallel Testing is being conducted in a polling place, or to activate it when there is no Parallel Testing being conducted.

b. *Detecting the Test Environment:* The Trojan Horse may determine that the machine is being parallel tested by looking at details of interactions between the machine and the "voters" or the larger voting system. For instance, many machines require the insertion of a smartcard or memory token before they would record any votes. Usually, there are several tokens in one polling place, and any one of them can be used on any machine. This means that most machines would have to scan a number of tokens during an Election Day. A Trojan Horse might be able to detect that a machine is being parallel tested if the same smartcard or memory token is inserted in that same machine all day.[134]

c. *Recognizing Vote Patterns:* The Trojan Horse may determine that the machine is being parallel tested by looking at the pattern of votes cast. For example, if Parallel Testing is done with randomly selected votes on each ballot position, the attack program may quickly notice that the Green and Libertarian candidates are getting about as many votes as the Republican and Democrat.

d. *Recognizing Usage Patterns:* The Trojan Horse may determine that the machine is being parallel tested by looking at usage patterns such as number of votes, speed of voting, time between voters, commonness of unusual requests like alternative languages or assistive technology, *etc.* For example, if Parallel Testing is always done for exactly 100 votes, the attack program may simply leave the vote totals unchanged when it gets exactly 100 votes.[135] At the 101st vote, the Trojan Horse knows it is safe to activate, and can switch vote totals at that point.

The first two of these are largely solvable by technical measures, the second two are impossible to solve in principle; they lead to a kind of arms race between attackers and defenders, and it is unclear where the ultimate advantage would lie.

Warning the Trojan Horse

There are many ways for attackers to (a) warn a Trojan Horse that its machine is being parallel tested, so that the Trojan Horse suppresses any attack behavior, or (b) activate Trojan Horses only on machines the attackers know are not being tested.

First, parallel testers, voters or poll workers could manually "tell" machines whether they were being tested. The attackers could:

- Send voters to each machine to be used in the attack with some kind of "Cryptic Knock" such as a voting pattern extremely unlikely to occur in the Parallel Testing.

  There are problems with this attack: voters sent to these machines need not be fully informed about the attack; however, since they know the Cryptic Knock, they could inform the authorities about what they have been told to do, and thus could allow parallel testers to detect the attack. Also, this strategy would require involving over 1,000 registered voters if this attack is expected to swing the governor's race in Pennasota.[136]

- Have election officials turn on each machine for voting using some Cryptic Knock to activate the attack program.

  This still requires a great many insiders in the attack team, since some election official or poll worker must do this for each machine to be activated. We have estimated that a minimum of 606 polling stations would need to be attacked in our state to swing an election, so this would involve the participation of 606 pollworkers.[137]

- Have parallel testers warn machine being tested with a Cryptic Knock. Because it would only take a single tester to warn the machine (and because the Cryptic Knock could be an innocuous action, like tapping the screen three times in the upper left hand corner), it would be unnecessary to subvert the entire team; instead it would only be necessary to subvert one team member per testing team, for a total of 58 testers.

In some circumstances, attackers could also warn the machines remotely:

- If the voting machine is capable of receiving wireless data transmissions, *e.g.*, over a wireless network or through an infrared port, then a single member of the attack team with a wireless-enabled PDA could activate or deactivate an entire polling place full of machines' attack programs, after verifying that no Parallel Testing is being done nearby. Thus, in jurisdictions that use machines capable of receiving wireless transmissions, an attack with enough foresight and technical skills could subvert Parallel Testing by increasing her attack team size by a relatively small number (and none of these attackers would need to be insiders).[138]

- If the voting machine is networked with other voting machines, and the attack program is installed in all of them, then a voter or election official need only use her "Cryptic Knock" once per polling place to enable or disable the attack program on all connected machines.

By removing any wireless channel by which voting machines can receive commands, the designer of a voting system can make Parallel Testing much more effective.

- If the voting machine is connected over some network to a central server, an attacker could issue a centralized warning telling compromised machines at polling places that would be receiving Parallel Testing teams not to trigger their attack behavior. This would require the attacker to (1) learn where Parallel Testing was occurring, and (2) find a method for infiltrating the network. Given the fact that jurisdictions in California are told which polling place would be tested at least a month in advance, it is plausible to imagine that this attack (against machines connected by a network to central servers) could be successful without dramatically increasing attack team size.

This discussion demonstrates the importance of closing machines off to remote communications: by removing any wireless channel by which voting machines can receive commands, the designer of a voting system can make Parallel Testing much more effective. Similarly, the effectiveness of Parallel Testing can be improved if the machines are never connected to one another during voting. Ideally, a voting machine would be completely isolated during the election, and would print out or otherwise display its totals *before* being connected to any central server to send in its tallies.

Detecting the Test Environment

The voting machine operates in a certain environment, in terms of its inputs and outputs. Some aspects of this environment may yield clues that it is being tested, rather than used in a normal election. For example:

- If the machine is treated in some different way when being activated for testing rather than normal voting, this is an easy thing to detect. For example, if a machine has its usual memory card replaced just before Parallel Testing, but it does not under normal circumstances, the attack program could use this to determine that it is being tested.

- Many machines require entry of authorization codes, memory tokens, or smartcards before a vote can be cast. If the same memory token/smartcard is presented many times, or if sequentially issued authorization codes are used for the voting machine, this could indicate testing.

An attacker's knowledge of these idiosyncrasies in the testing process could allow her to defeat Parallel Testing without increasing the attack team size. As long as there is something detectable by the machine that *always* occurs in Parallel Testing, the attacker should be able to defeat it. Take, for example, the situation where authorization codes are entered in tested machines sequentially. If the attacker knows that this always occurs in Parallel Testing, she simply instructs the Trojan Horse to shut off in the event that the first five authorization codes are entered sequentially. The testers would see no evidence that a Trojan Horse has been inserted into the machine.

To the furthest extent possible, the voting machine would be designed so that Parallel Testing would look no different from normal voting to the machine. More importantly, the precise techniques used for Parallel Testing (*e.g.,* exactly how and when the machine is activated, how activation codes/smart cards/*etc.* are produced to allow voting, *etc.*) should not be fully determined or revealed until immediately before the election. For Parallel Testing to be most effective, details of how Parallel Testing is conducted would change from election to election.

Recognizing Voting Patterns

Parallel Testing means voting on the machine to verify that it is accurately collecting votes. However, some voting patterns are more likely to occur than others, and some voting patterns are almost certainly artificial. If a Trojan Horse recognizes a voting pattern as artificial, it can shut off and defeat Parallel Testing

There are at least three ways for an attacker to exploit this vulnerability:

- The attacker could help write the Parallel Testing script, or have some power over it. She could use this to ensure that its outputs are always detected by the attack program. For example, the attacker could design the Parallel Testing script to undervote the first race on the 40th and 50th ballots cast; the attack program would notice this and refrain from any attack behavior when it happens.[139]

- The attacker could analyze the program or process used to generate the scripts (sequence of votes) used for Parallel Testing, and find a way to detect these scripts. For example, if the Parallel Testing program always generates a script where the first vote is one for a Whig, the attack program might only trigger where the first vote is for a non-Whig candidate.

- The attacker could analyze the statistical pattern of actual ballots cast in the state or county she is attacking, and attempt to design her program to catch suspicious patterns. This involves a kind of arms race between the attacker and the designer of Parallel Testing scripts. It also involves a huge increase in the sophistication of the attack program.

An important lesson from this analysis is that, to make Parallel Testing more effective, the Parallel Testing scripts need to be generated in a way that is not completely under any one person's control. The program used to generate Parallel Testing scripts would be kept secret until the day of the election; afterward, it would be fully specified, along with the random number generator seed used to produce the scripts, so that everyone can see that the scripts were generated in a sensible way and according to the program. It is important to design the Parallel Testing scripts so that they do not deviate in obvious ways from real voting patterns: that is, to be most effective, the scripts would include occasional third-party votes, undervotes, straight-ticket votes, and would broadly track likely voting patterns.

If an attacker can develop a Trojan Horse that can separate real voting patterns from Parallel Testing patterns, she can subvert Parallel Testing without increasing her attack team size. On the other hand, it is important to recognize the difference between a theoretical difference in voting patterns and one that is useful to an attack program. The attack program must run on voting machines in a wide variety of polling places, with a wide variety of voting patterns. Refusing to activate the attack behavior for too many voting patterns would leave the attack program unable to operate, and the attacker has a daunting data analysis problem to solve before she can determine how to write her attack program.

Recognizing Usage Patterns

Finally, the attack program may recognize usage patterns from the parallel testers that differ in important ways from those of real voters. For example:

- Parallel testers are likely to vote much more quickly and efficiently than most voters, especially after having cast many previous votes.

- A team of one or two parallel testers using the machine would not vary much in their usage pattern; that is, there would not be a fast, efficient voter followed by a rather slow one as one would expect to see in a real election.

- Parallel testers are likely to have very different patterns of errors (*e.g.*, attempted overvotes, undervotes to which they return and correct) than normal voters, especially after casting many votes.

- Parallel testers may take less time between casting ballots than normal voters; in a real election, the pattern of voters coming to vote would vary during the day.

- Unless instructed to do so by the script, most parallel testers are not likely to use assistive technology or alternative languages. On the other hand, if Parallel Testing scripts always include at least one use of each alternative language, this is likely to allow the attack program to determine that it is being tested.

One lesson from this analysis is that, to be most effective, the Parallel Testing script should include details like how quickly or slowly to vote, when to make "errors," and perhaps even when to cast each vote. [140]

More generally, our review of Parallel Testing leads us to conclude that good Parallel Testing should make a successful Software Attack Program significantly more complex to execute successfully. In terms of forcing our attacker to add additional informed participants to her attack, it might only require the addition

Good Parallel Testing should make a successful Software Attack Program significantly more complex to execute successfully.

of one to three people. This could be someone in control of writing, or with access to, Parallel Testing scripts. If such persons worked in conjunction with the designer of the Trojan Horse, they would have a good chance of subverting Parallel Testing. Similarly, conspirators with excellent knowledge of Parallel Testing procedures and practices could assist in the development of a Trojan Horse that could shut off when testing was detected.

TAKING ACTION WHEN PARALLEL TESTING FINDS DISCREPANCIES

Parallel Testing provides another problem: what happens when the electronic results reported by the machine do not match the script? In California, the process is relatively straightforward: a videotape of the testing is reviewed. The testers and Parallel Testing project manager examine the tape to determine whether human error (*i.e.*, where the tester has accidentally diverged from the script) is the cause of the discrepancy.[141]

If human error cannot explain the discrepancy, the Secretary of State's office impounds the machine and attempts to determine the source of the problem. Beyond this, even California does not appear to have a clear protocol in place.[142]

We have concluded that even if Parallel Testing reveals evidence of software bugs and/or attack programs on a voting machine, this countermeasure itself will be of questionable value unless jurisdictions have in place and adhere to effective policies and procedures for investigating such evidence, and taking remedial action, where appropriate. Detection of fraud without an appropriate response will not prevent attacks from succeeding. We offer an example of procedures that could allow jurisdictions to respond effectively to detection of bugs or software programs in Appendix M.

Adhering to such procedures when discrepancies are discovered during Parallel Testing is of the utmost importance. The misrecording of a single vote during Parallel Testing could indicate much wider problems.[143] Our analysis shows that Parallel Testing is a meaningful countermeasure only if there is a clear commitment to following investigative and remedial procedures when problems are discovered.

CONCLUSIONS AND OBSERVATIONS

Conclusions from the Representative Least Difficult Attack

With the Basic Set of Countermeasures in place, a minimum of one informed participant will be needed to successfully execute DRE Attack #4 (Trojan Horse Inserted Into Operating System) and change the result of the Pennasota Governor's race.

With the Basic Set of Countermeasures plus the Parallel Testing Regimen, DRE

Attack # 4 becomes more difficult. The attacker will need at least 2 to 4 informed participants[144] to successfully execute DRE Attack #4 and change the result of the Pennasota governor's race.

We are unable to examine whether the Basic Set of Countermeasures plus the Automatic Routine Audit Regimen would make DRE Attack #4 more difficult because DREs do not have a voter-verified paper trail.

Conclusions about Trojan Horse and other Software Attack Programs

- The Trojan Horse and other corrupt software attacks are extremely dangerous because they require very few (if any) co-conspirators and can affect enough votes to change the outcome of a statewide race.

- The Basic Set of Countermeasures currently used in many jurisdictions is not likely to catch a clever Trojan Horse or other Software Attack Program.

Conclusions about the Potential Effectiveness of Parallel Testing

- Parallel Testing, if conducted properly, will force an attacker who employs a Software Attack Program to spend much more time, and gain significant knowledge before she can execute a successful attack.

- Parallel Testing creates a kind of arms race between attackers and defenders: as Parallel Testing becomes more sophisticated, the attacker must become more sophisticated; as the attacker becomes more sophisticated, Parallel Testing must come up with new ways to trip her up. The single biggest problem with Parallel Testing is that, given the potential resources and motivation of an attacker, it is ultimately unclear whether the final advantage would lie with the testers or the attacker. Moreover, because Parallel Testing does not create an independent record of voters' choices, there is no reliable way to know whether an attack has successfully defeated Parallel Testing.

- Parallel Testing would not necessarily require an attacker to involve significantly more co-conspirators to employ her attack successfully. We have envisioned scenarios where the attacker could involve as few as one to three additional conspirators to circumvent Parallel Testing. Because of the "arms race" created by Parallel Testing, it is extremely difficult to assign a minimum number of attackers that might be needed to circumvent it.

Conclusions about Taking Action When Attacks or Bugs Are Discovered by Parallel Testing

- Parallel Testing as a countermeasure is of questionable value unless jurisdictions have in place and adhere to effective policies and procedures for investigating evidence of computer Software Attack Programs or bugs, and taking remedial action, where appropriate.

Key Observations about Parallel Testing

Our examination of Parallel Testing shows that the following techniques could make a Parallel Testing regime significantly more effective:

- The precise techniques used for Parallel Testing are not fully determined or revealed, even to the testers, until right before the election. Details of how Parallel Testing is conducted are changed from election to election.

- The wireless channels for voting machines to receive commands are closed.

- Voting machines are never connected to one another during voting. If they are normally connected, a voter or pollworker might be able to activate or deactivate a Trojan Horse on every machine in the polling place with one triggering command or event.

- Each voting machine is completely isolated during the election. This would prevent remote attacks from activating or deactivating the Trojan Horse.

- To the extent possible, the voting machines are designed so that Parallel Testing would look no different from real voting to the machine. Parallel Testing scripts could include details like how quickly or slowly to vote, when to make "errors," and perhaps even when to cast each vote.

Parallel Testing is videotaped to ensure that a contradiction between the script and machine records when Parallel Testing is complete is not the result of tester error.

ATTACKS AGAINST DREs W/ VVPT

We have identified over forty (40) potential attacks against DREs with VVPT.[145]As it was for DREs without VVPT, all of the least difficult attacks against DREs with VVPT involve inserting Trojan Horses or corrupt software into the DREs. The key difference in attacks against DREs with VVP T is that our attacker may also have to attack the paper trail.

A paper trail by itself would not necessarily make an attack on DREs more difficult. An attacker against DREs with VVPT has two options:

1. Ignore the paper trail in the attack. Under this scenario, only the electronic record of votes is targeted. The attacker hopes that the electronic record becomes the official record, and that no attempt is made to count the paper record, or to reconcile the paper and electronic records; or

2. Attack both the paper and electronic record. Under this scenario, the attacker would program her software record to change both the electronic and paper records. This attack would only work if a certain percentage of voters does not

review the paper record and notice that their votes have not been recorded correctly.

In this section, we examine examples of both types of attacks. Further, we evaluate how difficult each of these attacks would become if a jurisdiction implemented the "Basic," "PT Plus Basic," and "ARA Plus Basic" sets of countermeasures.

REPRESENTATIVE "LEAST DIFFICULT" ATTACK: TROJAN HORSE TRIGGERED WITH HIDDEN COMMANDS IN BALLOT DEFINITION FILE (DRE W/ VVPT ATTACK NO. 1A)

We have already discussed how a Trojan Horse might be inserted into a DRE. The insertion of a Software Attack Program into a DRE w/ VVPT would not differ in any significant way. It could be inserted into the software or firmware at the vendor, into the operating system, COTS software, patches and updates, *etc.* In most cases, this would require the involvement of a minimum of one attacker.

As already discussed (*see supra* p. 55), if the attacker wanted to tailor her attacks to specific precincts, she might create an attack program that would not activate unless it has been triggered. In this scenario, the attack would be "parameterized" (*i.e.,* told which ballot, precinct, race, *etc.* to attack) by commands that are fed into the machine at a later time. This allows the attacker to trigger an attack with specific instructions whenever she decides it could be useful.

Voting machine security experts sometimes imagine this triggering and parameterization would happen via the ballot definition files.[146] Ballot definition files tell the machine how to (1) display the races and candidates, and (2) record the votes cast. Ballot definition files are often written by the voting machine vendor employees or consultants, but they are also frequently written by local jurisdictions themselves (at the county level), with software and assistance provided by the vendor.[147]

A seemingly innocuous entry on the ballot definition file could be used to trigger the attack program. For instance, as already discussed, an extra space after the last name of a candidate for a particular race could trigger an attack that would subtract five votes from that candidate's total on every machine. This triggering is referred to as "parameterization" because it allows the attacker to set the parameters of the attack – *i.e.,* the ballot, the precinct (because there is a different ballot definition file for each precinct), the race, and the candidate who is affected.

If the vendor writes the ballot definition files for many counties in a state, only one person would be needed to trigger and parameterize the attack in many polling places.

If the vendor writes the ballot definition files for many counties in a state, only one person would be needed to trigger and parameterize the attack in many polling places.

This attack would become more difficult if every county created its own ballot definition file. In such cases, the attacker would have to find one participant per county to help her with her attack. In addition to forcing the attacker to expand the number of participants working with her, creating the ballot definition files locally could force the attackers to infiltrate the election offices of multiple counties.

Here is how this representative attack could happen in Pennasota:[148]

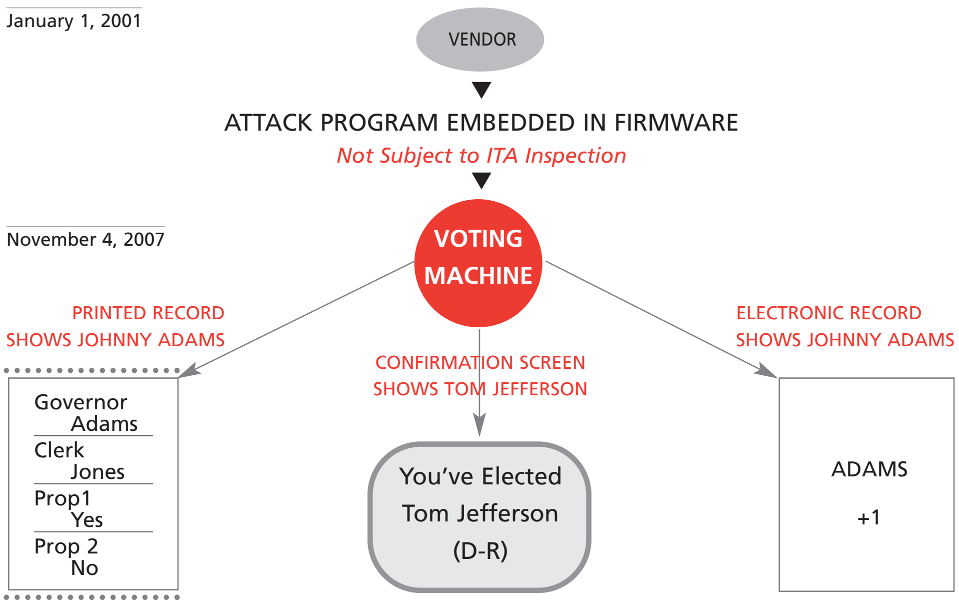
- The Software Attack Program is created and inserted at any time prior to an election.
- If the ballot definition files are created at the vendor, or by a consultant provided by the vendor: Someone at the vendor involved in creating, editing or reviewing the ballot definition files would insert the commands that tell the Attack Program which race to target.
- If the ballot definition files are created by local jurisdictions: three separate people working in the election offices of the three largest counties insert commands into the ballot definition files. Obviously, these co-conspirators would have to possess access to the ballot definition files.
  - The Software Attack Program could be set to activate on a specific date and time (*e.g.*, the first Tuesday after the first Monday in November, after 11 a.m.). This would help it avoid detection during Logic and Accuracy testing; there would be no need to worry about ITA or acceptance testing, as the ballot definition file is not subjected to either of these tests.
  - When switching votes, the ballot definition file could show voters Tom Jefferson on the confirmation screen, but electronically record a vote for Johnny Adams.
  - Alternatively, the Software Attack Program could alter votes in the electronic memory at the end of a full day of voting.
- To avoid detection after the polls have closed, the Software Attack Program could create and provide a new log to the outside world, different than the one stored internally.

In the gubernatorial election for the State of Pennasota, we have calculated that if a Trojan Horse were inserted into the ballot definition files for *only* the three largest counties, it would need only to switch four (4) votes per machine (or less than 5% of votes per machine) to change the results of our close statewide election:

- Total votes Johnny Adams needs to switch for comfortable victory: 51,891
- Number of DREs w/ VVPT in 3 largest counties: 9,634[149]

FIGURE 7

POSSIBLE ATTACK ON DRE WITH VVPT


January 1, 2001
VENDOR
ATTACK PROGRAM EMBEDDED IN FIRMWARE
Not Subject to ITA Inspection
November 4, 2007
VOTING MACHINE
PRINTED RECORD SHOWS JOHNNY ADAMS
CONFIRMATION SCREEN SHOWS TOM JEFFERSON
ELECTRONIC RECORD SHOWS JOHNNY ADAMS
Governor Adams
Clerk Jones
Prop1 Yes
Prop 2 No
You've Elected Tom Jefferson (D-R)
ADAMS
+1

- If four (4) votes on each machine in the three largest counties were switched, Johnny Adams would have gained enough votes to defeat Tom Jefferson comfortably.

Thus, this attack would require between two and four participants: one to insert the Software Attack Program, plus either one or three (depending upon whether ballot definition files were created at the vendor or county) to provide triggering and parameterization commands in the ballot definition files.

Although it might be more difficult than other types of Trojan Horse attacks (because it could require one informed participant per county, as opposed to a single informed participant via several points of entry), the "Trojan Horse Triggered by Hidden Commands in the Ballot Definition File" attack has certain elements that would render it less difficult to execute:

- This attack provides the attackers a great deal of flexibility. The attackers can wait until just before any election to trigger an attack, and their attack can target specific precincts; and

- This attack is reusable. The attack program would not do anything unless it receives commands from ballot definition files. These commands could come before any election and the attack program could lie dormant and undetected for many election cycles.

ATTACKING BOTH PAPER AND ELECTRONIC RECORDS (DRE W/ VVPT ATTACK NO. 6)

In the above analysis, we assumed that the paper trail is not attacked: only the electronic record misrecorded the vote. Would not this mean that the attack would be detected? Not necessarily.

Even in states with mandatory voter-verified paper trails, official vote totals are still extracted from the electronic record of the machine. While an attacker might have to worry that a VVPT recount in a close race would expose the attack, statewide recounts are still relatively rare.[150]

PAPER MISRECORDS VOTE

To prevent an attack from being noticed in a recount, our attacker could create a Software Attack Program that also directs the printer to record the wrong vote. This "Paper Misrecords Vote" attack is Attack Number 6 in the DRE w/ VVPT Catalog.

The attack could work the same way as DRE w/ VVPT Attack Number 1a (Trojan Horse Triggered with Hidden Commands in Ballot Definition File),[151] except that it would add a step: the paper receipt printed after the voter has made all of her selections would incorrectly record her vote for governor. In practice, this is how it would work:

- When a targeted voter chooses Tom Jefferson, the screen would indicate that she has voted for Tom Jefferson.
- After she has completed voting in all other races, the DRE would print a paper record that lists her choices for every race, except for governor. Under the governor's race, it would state that she has selected Johnny Adams.
- When the DRE screen asks the voter to confirm that the paper has recorded her vote correctly, one of two things would happen:
  - the voter would fail to notice that the paper has misrecorded the vote and accept the paper recording; or
  - the voter would reject the paper record, and opt to vote again.
- If the voter rejects the paper record, the second time around it would show that she voted for Tom Jefferson. This might lead her to believe she had accidentally pressed the wrong candidate the first time. In any event, it might make her less likely to tell anyone that the machine made a mistake.

This attack would not require any additional participants in the conspiracy. Nor is it entirely clear that enough voters would notice the misrecorded votes to prevent the attack from working.

DO VOTERS REVIEW VVPT?

In a recent study, Professor Ted Selker and Sharon Cohen of MIT paid 36 subjects to vote on DRE w/ VVPT machines.[152] They reported that "[o]ut of 108 elections that contained errors . . . only 3 [errors were recognized] while using the VVPT system."[153]

If only 3 of every 108 voters noticed when the paper trail misrecorded a vote for Tom Jefferson as a vote for Johnny Adams, DRE w/ VVPT Attack Number 6 would probably work. If the Trojan Horse targeted approximately 54,000 voters for Tom Jefferson (or roughly 1 in every 9 voters for Tom Jefferson in the three largest counties), the vast majority would not notice that the paper had misrecorded their votes. 3% – or 1,633 – would notice. These voters would cancel the paper record and vote again. The second time, the paper would record their votes correctly.

FIGURE 8

51,891	Total votes Johnny Adams needs to switch for comfortable victory
3,459,379	Total votes
54,437	Votes attacked
3.0%	% of voters who study VVPT carefully
1,633	number of rejections of misrecorded votes
52,804	number of votes successfully switched

This would still leave enough switched votes for Johnny Adams to win the governor's race comfortably. We do not know how many of the 1,633 voters who rejected their votes would complain to poll workers that the machines had initially misrecorded their votes. But even if 50% of those voters were to complain,[154] this would be an exceptionally small number of complaints. With nearly 1,700 precincts and 10,000 DREs w/ VVPT in the three largest counties, 820 complaints amount to less than one complaint per two precincts and twelve machines.[155]

We are skeptical that in the State of Pennasota, only 3% of voters would notice if their choice for governor was misrecorded on the paper trail. This is because (1) the race that we are looking at is for the top office in the state; this is an election with which voters are more likely to be concerned and, consequently, they would be more likely to check that the VVPT has correctly recorded their votes (as opposed to their votes for, say Proposition 42, which is likely to be in the middle or bottom of their paper trail), and (2) in an actual election (as opposed to the MIT study), where candidates should be well known to most voters, they are probably more likely to notice if the paper trail accurately reflects their choice.

Convincing voters to review their VVPT is critical to its effectiveness as a measure to thwart certain Trojan Horse attacks.

Keeping in mind that the attacker's goal is to switch 51,891 votes, let us assume that 20% of all voters for Tom Jefferson in our three targeted counties would check to see that the paper has accurately recorded their votes. The attacker could reach her goal by targeting 66,000 voters for Tom Jefferson (out of nearly 1.1 million votes cast in these counties). Over 13,200 of these voters would notice that the paper misrecorded their choice; they would recast their votes. But over 52,800 would not notice; these extra 52,800 votes would be sufficient to change the outcome of the election.

FIGURE 9

51,891	Total votes Johnny Adams needs to switch for comfortable victory
3,459,379	Total votes
66,004	Votes attacked
20.0%	% of voters who study VVPT carefully
13,201	number of rejections of misrecorded votes
52,804	number of votes successfully switched

It might be argued that if 13,200 people noticed that their votes had been misrecorded on the VVPT, someone would realize that something was wrong with the machines. The truth is, we cannot know what would happen if this number of people were to notice that their votes were misrecorded. As already discussed, many people would probably presume that the mistake was theirs, and not that of the machine.

By contrast, if 80% of voters for Tom Jefferson in the three counties checked their paper records thoroughly, it is doubtful the attack could succeed. The Trojan Horse would have to target over 264,000 voters for Tom Jefferson to get the 51,891 needed to ensure victory for Johnny Adams. 211,212 voters for Tom Jefferson would notice that the paper trail initially recorded their votes incorrectly; this represents over 40% of all of his votes in the three largest counties.

We can see from this analysis that convincing voters to review their VVPT is critical to its effectiveness as a measure to thwart certain Trojan Horse attacks.

The Trojan Horse could be programmed in a way that would allow it to detect whether it is being tested.

THE EFFECT OF REGIMEN FOR PARALLEL TESTING PLUS BASIC SET OF COUNTERMEASURES

Our analysis of the effect of the Basic and Regimen for Parallel Testing Plus Basic Sets of Countermeasures against the least difficult attack for DREs with VVPT does not dramatically change from the same analysis done for DREs without VVPT. Unless voters check the paper trail and report suspected misrecordings to poll workers when they occur, the paper trail, by itself, provides very little additional security.

The Regimen for Parallel Testing plus Basic Set of Countermeasures should provide more protection than just the Basic Set of Countermeasures. In fact, if the Software Attack Program does not recognize that it is being tested, Parallel Testing would probably catch this type of attack; presumably at least one tester would notice that the paper record was not recording correctly.

However, as already discussed, *supra*, we have concerns about certain vulnerabilities in Parallel Testing: first, there is the possibility that the person installing the ballot definition file commands triggering the attack program would know which precincts are going to be subject to Parallel Testing – in California, precincts are told at least one month in advance whether their machines would be tested.[156] If the attacker knew where the Parallel Testing is going to occur, she could simply refrain from inserting the triggering commands in ballot definition files for those precincts.

Second, the attacker could, via a wireless communication or Cryptic Knock (1) activate the Trojan Horse on machines she sees are not being tested on Election Day, or (2) de-activate the Trojan Horse on machines she sees are being tested on Election Day (this presumes that Parallel Testing is done at the polling stations).

Finally, the Trojan Horse could have been programmed in a way that would allow it to detect whether it is being tested: if the attacker knew something about the testing script in advance, or had a good understanding of Parallel Testing procedures, she might be able to program the Trojan Horse to shut off during all Parallel Testing.

As already discussed, the successful subversion of Parallel Testing, while adding significant complexity to a software attack, might require the additional participation of between only one and three extra informed participants.

EFFECT OF REGIMEN FOR AUTOMATIC ROUTINE AUDIT PLUS BASIC SET OF COUNTERMEASURES

The Regimen for Automatic Routine Audit plus Basic Set of Countermeasures, if instituted as detailed *infra* at pp. 16–17, should be an effective countermeasure against our least difficult attack. As detailed in Appendix K, if 2% of all machines were audited, auditors should have a greater than 95% chance of discovering a mismatch between electronic records and paper records, where a Trojan Horse misrecorded a voter's choice in the paper record. This, of course, presumes that the attacker failed to find a way to subvert the Regimen for Automatic Routine Audit.

We have identified at least four ways an attacker could subvert the Regimen for Automatic Routine Audit:

a. The Trojan Horse attacks both paper and electronic records and most voters do not review the paper record before casting their votes, resulting in an attack that successfully subverts both the electronic and paper record;

b. The selection of auditors is co-opted;

c. The paper record is replaced before an audit of the voter-verified paper record takes place, for the purpose of matching paper records to corrupted electronic records; or

d. The paper record is replaced merely to add votes for one candidate, without regard to what has occurred in electronic record.

As with our analysis of the Regimen for Parallel Testing, to determine the likely effectiveness of the Regimen for Automatic Routine Audit, we must ask how much more difficult it would make our least difficult attack. This means, among other things, examining how many people it would take to subvert the Regimen for Automatic Routine Audit by each of the three methods listed above.

TROJAN HORSE ATTACKS PAPER AT TIME OF VOTING, VOTERS FAIL TO REVIEW

Our attacker does not necessarily need to attack the audit process directly to subvert it. What if, as already described in our discussion of DRE w/ VVPT Attack Number 6 (*see supra* at p. 65), the attacker merely designs a Trojan Horse that changes both the paper and electronic record?

As noted above, if 80% of voters thoroughly reviewed their paper trails, it is very likely that an attack on the paper trail at the time of voting would fail. Assuming, however, that this attack is noticed by voters for Tom Jefferson only 20% of the time, how much more difficult would the Regimen for Automatic Routine Audit make the attack?

If the audit of the voter-verified paper record merely adds up total votes on paper and compares them to total votes in the electronic record, it is doubtful this attack would be discovered by election officials. The paper record would match the electronic record. The attacker would not need to add any people to her conspiracy

Jurisdictions will have to put in place certain rules regarding what is to be done when anomalies are found.

to succeed.

If, on the other hand, the audit of the voter-verified paper record looks for statistical anomalies by, for instance, looking at the number of times voters cancelled the paper record of their vote, this attack is likely to be caught. As already noted, if 20% of targeted voters notice that their paper record has not correctly recorded their vote for Tom Jefferson, there would be more than 9,500 cancellations showing Johnny Adams' name crossed out, and subsequently replaced by Tom Jefferson:

FIGURE 10

51,891	Total votes Johnny Adams needs to switch for comfortable victory
3,459,379	Total votes
66,004	Votes attacked
20.0%	% of voters who study VVPT carefully
13,201	number of rejections of misrecorded votes
52,803	number of votes successfully switched

13,201 votes is an extremely small percentage of the 3.4 million votes cast. However, it would represent an unusually large number of cancellations. Larry Lomax, Registrar of Voters for Clark County, Nevada (which has used DREs w/ VVPT since 2004) states that in Clark County it is "the exception" to find a single cancellation on a DRE's entire roll of paper trail.[157] Even if we were to assume that it is normal to have one cancellation for every two DREs w/ VVPT, this would mean that in Pennasota, there would ordinarily be about 14,000-15,000 cancellations in the entire state.[158] Thus, an audit of the voter-verified paper record that looked for statistical anomalies like cancellations would show that there were 90% more cancellations than normal.

An audit of the voter-verified paper record that noted which votes were changed after cancellation would show an even more troubling pattern: a highly disproportionate number of cancellations where the paper record changed from Johnny Adams to Tom Jefferson.

Finally, to the extent this attack is limited to the smallest possible number of polling places in three counties (as we originally suggested), certain audits would show an even higher statistical anomaly – with an additional 22 paper cancellations per polling place.[159]

Of course, finding statistical anomalies, no matter how troubling, would not, *in and of itself,* thwart an attack. Jurisdictions will have to put in place certain rules regarding what is to be done when such anomalies are found.

Other than requiring auditors and election officials to look for discrepancies between paper and electronic records, states do not currently mandate review of paper records for statistical anomalies. States that do not review statistical anomalies (such as, for instance, an unusually high number of cancellations or skipped races) during audit will remain vulnerable to a number of attacks.

Our analysis shows that unless a jurisdiction implements and adheres to effective policies and procedures for investigating such anomalies (and taking remedial action, where appropriate), a review of statistical anomalies will be of questionable security value. We provide examples of procedures that would allow jurisdictions to respond effectively to detection of statistical anomalies in the voter-verified paper record in Appendix J.

CO-OPTING THE AUDITORS

An obvious, but difficult way to subvert the audit is to directly co-opt the auditors. However, given the fact that under the Automatic Routine Audit Regimen, audit teams are randomly assigned to randomly selected voting machines, it would be exceptionally difficult to defeat the Regimen for Automatic Routine Audit by co-opting the auditors. We have estimated that in an audit of 2% of all machines, there would be 386 auditors randomly assigned to machines in the three largest counties in Pennasota.[160]As demonstrated in Appendix L, to have a reasonable chance of subverting the audit by infiltrating the auditors, it would be necessary to subvert all of them.

Of course, if a corrupt person selects the auditors or polling places and does not follow the "transparent random selection process" discussed *supra* at p. 17, subversion of the Automatic Routine Audit becomes much easier. For instance, if the attacker were in control of the decision as to which polling places to pick for the audit, she could deliberately choose those polling places that she knows the Trojan Horse did not attack. For this reason, transparent randomness (as discussed in detail, *see infra* at pp. 42–113) is critical to an effective audit.

REPLACING PAPER BEFORE THE AUTOMATIC ROUTINE AUDIT TAKES PLACE

Another way to subvert the Regimen for Automatic Routine Audit is to replace the paper before an audit can be completed, for the purpose of making sure that the audited paper records match the corrupted electronic records. This would be nearly impossible if the audit of the voter-verified paper record was conducted in the polling places immediately after the polls close.

We understand that for many jurisdictions, this will not be realistic. After spending all day at the polls, it is likely that pollworkers and election officials would not want to spend additional time assisting auditors as they conduct an audit of the voter-verified paper record. Moreover, many audit volunteers may be reluctant to begin conducting an audit (which would, at the very least, take several hours) at 9 or 10 p.m.

If the audit of the voter-verified paper record is not conducted at the polls immediately upon their closing, there are at least two ways in which an attacker could corrupt or replace the paper trail: (1) by intercepting and replacing the paper while it is in transit to the warehouse or county offices where the audit would take place, or (2) by replacing the paper where it is stored prior to the audit.

If there are very strong physical security measures, such as those assumed in the Basic Set of Countermeasures, and paper from each polling place is delivered to the audit location separately, task (1) would be extremely difficult. Even assuming the attackers have attacked the minimum number of polling places (606), they would need to intercept and replace more than 550 separate convoys of paper to have even a one in three chance the audit would not catch the fact that some paper record had different totals than the electronic record.[161] Given that, in most states, all polls close at the same time, this would seem to require the participation of at least 1,100 additional informed participants – making the attack far more difficult.

The other alternative would be to attempt to replace the paper records at the county warehouses, prior to the audit. As already discussed, our assumption is that our attackers would need to target a minimum of three counties to change the outcome of the governor's race in Pennasota. This means, at a minimum, that our attackers would need to target three separate county warehouses and replace the paper records stored there.

Again, if very strong physical security measures and the chain of custody practices assumed in the Basic Set of Countermeasures are followed, this should be very difficult.

We have estimated that 2,883 DREs w/ VVPT would have to be replaced to change the outcome of a statewide race.[162] In Pennasaota, the voter-verified paper records of each of these machines would have been sealed with tamper evident seals and stored in a room with perimeter alarms, secure locks, video surveillance, and there would be regular visits by security guards and police officers. The seal numbers would have been assigned at the polling place and logged by county officials upon reaching the county warehouse.

We have assumed that the audit of the voter-verified paper record would begin at 9 a.m. the morning after the polls closed, so our attackers would have to subvert all of these precautions and replace the paper trails for nearly 2,117 DREs w/ VVPT in three county warehouses within a matter of hours to ensure that the attack was not discovered during the audit.[163]

Aside from the fact that, in Pennasota, our attackers would (in this very short time period) need to (1) break and replace thousands of tamper evident seals in three separate locations,[164] (2) get past the warehouse locks and alarms, (3) co-opt (or avoid detection of) the randomly assigned police officers and security guards at each location,[165] and (4) somehow avoid detection from the video surveillance, the attackers would also need to deliver and replace 2,117 rolls of VVPT (or, in the case of PCOS, about 40,000 separate ballots) without independent observers outside or inside the warehouse noticing. We have concluded that it would not be feasible to carry out this attack without detection over such a short period of time, unless the attackers had the cooperation of hundreds of participants, including many insiders (*i.e.,* security guards, policemen and video-monitors).

REPLACING SOME PAPER RECORDS MERELY TO ADD VOTES

Our attackers have a final option: attack the paper records, not for the purpose of reconciling them with the electronic records, but merely to add enough paper votes to Adams's total to ensure that the paper records also show him winning. This would merely mean stuffing enough ballot boxes with additional ballots to give Adams a majority of votes in the paper record.

The audit of the voter-verified paper record would then show a discrepancy between the electronic and paper records. A recount would follow. It would show that Adams had more votes in the paper record. In 15 states, the VVPT laws specify that "if there is a recount, the paper ballot" is the official record.[166]

There are a number of problems associated with a bright line rule stating that the paper (or electronic) record will always control election results. There is certainly nothing wrong with providing that paper records will have a "presumption" of authority. A bright line rule, however, could invite the kind of deception we are seeking to prevent.

As this analysis shows, the main benefit of paper, when accompanied by the Regimen for Automatic Routine Audit, is that it requires the attackers to subvert *both* the electronic and paper records. If the attackers know that they only have to attack the paper record, their attack becomes significantly easier.

In our scenario, the attackers would successfully insert the Trojan Horse. Obviously, they would not have to do this if they knew the paper record always controlled. They could merely attack the paper record and hope the audit of the voter-verified paper record would spot a contradiction between the paper and electronic records (which it almost certainly would if they switched enough votes to change the outcome of the election).

But let us suppose they did insert the Trojan Horse. If they intercepted 60 convoys of paper (or merely replaced several ballot boxes in 60 polling places before they were transported), they could replace enough paper to create a victory for

Johnny Adams in the paper record as well.[167] While not easy, this attack is clearly much easier (involving at least 1,000 fewer participants) than one that would require the attackers to prevent the audit of the voter-verified paper record from revealing contradictory paper and electronic records.

Of course, when the audit of the voter-verified paper record was conducted, Pennasota would discover that something strange had happened: in at least a few audited polling places, the paper and electronic records would not match.

But this would not tell Pennasota who won. A recount would show Johnny Adams winning under either set of records. A bright line rule about which record should govern in such circumstances is problematic. It would encourage the kind of deception we have imagined in this attack: if Pennasota had a law stating paper records should govern (as provided in California),[168] Johnny Adams would win. If the law stated that electronic records govern (as provided in Idaho and Nevada),[169] Johnny Adams would still win.

What can be done to prevent this attack? We discuss this below.

TAKING ACTION WHEN AUTOMATIC ROUTINE AUDIT FINDS ANOMALIES

Many state statutes are silent as to what should happen when paper and electronic records cannot be reconciled. As already discussed, Illinois law provides that where electronic and paper records in the Automatic Routine Audit do not match, the county notifies "the State Board of Elections, the State's Attorney and other appropriate law enforcement agencies, the county leader of each political party and qualified civic organizations."[170]

As with Parallel Testing, an Automatic Routine Audit offers questionable security benefit unless effective procedures to investigate discrepancies (including taking remedial action, where necessary) are implemented and adhered to. Again, detection of possible fraud without an effective response will not thwart an attack on voting systems. The following are examples of procedures that would allow jurisdictions to respond effectively to discrepancies between paper and electronic records during an Automatic Routine Audit:

1. conduct a transparent investigation on all machines where the paper and electronic records do not match to determine whether there is any evidence that tampering with the paper records has occurred;[171]

2. to the extent that there is no record that the paper records have been tampered with, certify the paper records;

3. if there is evidence that the paper records have been tampered with, give a presumption of authority to the electronic records;

4. after giving a presumption of authority to the electronic records, conduct a forensic investigation on all machines where the paper and electronic records do not match. The purpose of this investigation would be to determine whether there has been any tampering with the electronic records;

5. if tampering with the electronic records can be ruled out, certify the electronic records;[172]

6. where there is evidence that both sets of records have been tampered with, conduct a full recount to determine whether and to what extent paper and electronic records cannot be reconciled;

7. at the conclusion of the full recount, determine the total number of machines that report different electronic and paper records;

8. after quantifying the number of machines that have been tampered with, determine the margin of victory in each potentially affected race;

9. based upon (a) the margin of victory, (b) the number of machines affected, and (c) the nature and scope of the tampering, determine whether there is a substantial likelihood that tampering change the outcome of a particular race; and

10. in the event that a determination is made that there is a substantial likelihood that tampering changed the outcome of a particular race, hold a new election for the office.

CONCLUSIONS

Conclusions from the Representative Least Difficult Attack

Assuming that only 20% of voters review their voter-verified paper trail, a minimum of one to three informed participants[173] will be needed to successfully execute DRE w/ VVPT Attack #6 (Memory and Paper Misrecord Vote Due to Trojan Horse Inserted in Ballot Definition File) and change the result of the Pennasota Governor's race.

Assuming that 80% of voters review their voter-verified paper trail, DRE w/ VVPT Attack #6 will not succeed.

With the Basic Set of Countermeasures plus the Parallel Testing Regimen, DRE w/ VVPT Attack # 6 becomes more difficult. The attacker will need at least 2 to 6 informed participants to successfully execute DRE w/ VVPT Attack #6 and change the result of the Pennasota governor's race.

DRE Attack w/ VVPT Attack #6 would be substantially more difficult to successfully execute against the Basic Set of Countermeasures Plus the Automatic Routine Audit Regimen than it would be against the Basic Set of Countermeasures or the Basic Set of Countermeasures plus the Parallel Testing Regimen. The attacker will need at least 386 informed participants to successfully execute DRE w/ VVPT Attack #6 and change the result of the Pennasota Governor's race.

Conclusions about the DRE w/ VVPT

- As with DREs without VVPT, local jurisdictions that take control of important tasks, like creating ballot definition files, will make successful statewide attacks more difficult.

- The value of paper without an Automatic Routine Audit against many attacks (such as DRE Attack #1a, where the electronic record is changed, but the paper record is not) is highly questionable.

- If voters are encouraged to review their VVPT thoroughly before casting their votes, many of the least difficult attacks against DREs w/ VVPT will become substantially more difficult.

Conclusions about the Regimen for Automatic Routine Audit Plus Basic Set of Countermeasures

- Statistical examination of anomalies, such as higher than expected cancellations, can help to detect fraud. Currently, none of the states that conduct routine audits of voter-verified paper records examine those paper records for statistical anomalies.

- Automatic Routine Audits conducted soon after the close of polls are less vulnerable to attack, because there is less time to tamper with the paper records.

- Good chain of custody practices and physical security of paper records prior to the Automatic Routine Audit is crucial to creating an effective auditing regimen. Specifically, the following practices should make the auditing process more secure:

  - At close of the polls, vote tallies for each machine are totaled and compared with number of persons that have signed the poll books.

  - A copy of totals for each machine is posted at each polling place on Election Night.

  - All audit information (*i.e.*, Event Logs, VVPT records, paper ballots, machine printouts of totals) that is not electronically transmitted as part of the unofficial upload to the central election office, is delivered in official, sealed and hand-delivered information packets or boxes. All seals are tamper-resistant.

- Transportation of information packets are completed by at least two election officials representing opposing parties who have been instructed to remain in joint custody of the information packets or boxes from the moment it leaves the precinct to the moment it arrives at the county election center.
- Each polling place sends its information packets or boxes to the county election center separately, rather than having one truck or person pick up this data from multiple polling locations.
- Once the sealed information packets or boxes have reached the county election center, they are logged. Numbers on the seals are checked to ensure that they have not been replaced. Any broken or replaced seals are logged. Intact seals are left intact by officials.
- After the packets and/or boxes have been logged, they are provided with physical security precautions at least as great as those listed for voting machines, above. Specifically: the room in which they are stored would have perimeter alarms, secure locks, video surveillance and regular visits by security guards; and access to the room is controlled by sign-in, possibly with card keys or similar automatic logging of entry and exit for regular staff.

- The auditing process will be much less vulnerable to attack if machines and auditors are selected and assigned in a publicly transparent manner.

An automatic routine audit offers questionable security benefit unless effective procedures to investigate discrepancies (including taking remedial action, where necessary) are consistently implemented.

Conclusions about Taking Action When Anomalies Are Found in the Automatic Routine Audit

An automatic routine audit offers questionable security benefit unless effective procedures to investigate discrepancies (including taking remedial action, where necessary) are consistently implemented. Detection of possible fraud without an effective response will not thwart an attack on voting systems.

ATTACKS AGAINST PCOS

We have identified over forty (40) potential attacks against PCOS. Many of these attacks are similar to the attacks against both DRE systems.

Nothing in our research or analysis has shown that a Trojan Horse or other Software Attack Programs would be more difficult against PCOS systems than they are against DREs. All of the least difficult attacks against PCOS involve the insertion of Trojan Horses or corrupt software into PCOS scanners.[174] In this section, we examine how this attack would work, and how much more "expensive" such attacks would be made by the "Basic," "Regimen for Parallel Testing Plus Basic" and "Regimen for Automatic Routine Audit Plus Basic" set of countermeasures.

We also address certain security concerns that are unique to the PCOS system.

REPRESENTATIVE "LEAST DIFFICULT" ATTACK: SOFTWARE ATTACK INSERTED ON MEMORY CARDS (PCOS ATTACK NO. 41)

We have already discussed how a Trojan Horse might be inserted into both DRE systems. The insertion of a Trojan Horse into a PCOS scanner would not differ in any significant way. It could be inserted into the main PCOS source code tree, operating system, COTS software, and software patches and updates, *etc. In most cases, this would require the involvement of a minimum of one person.*

Attack Number 41 in the PCOS Catalog is an attack that has been demonstrated to work in at least two election simulations:[175] use of memory cards to change the electronic results reported by the PCOS scanner. While this attack has only been publicly attempted against one model of PCOS scanner, several computer security experts that have reviewed other PCOS systems believe that they may be vulnerable to similar attacks.[176]

DESCRIPTION OF ATTACK

This attack uses replaceable memory cards to install the software attack. Memory cards are used by both DREs and PCOS scanners. Memory cards contain data that is used by the machines, including the ballot definition files (which allow the machine to read the ballots) and the vote totals. At least one major vendor has its report generation program on its memory cards: this is the program that, among other things, tells the machine what vote totals to print at the close of the polls. This is the record pollworkers use to record the final votes in each machine.

Attackers could use the memory cards to generate false vote total reports from the machine. Here is how the attack would work:

- The attacker acquires access to the memory cards before they are sent to individual polling places. She could gain access
  - at the county office where they are programmed, if she works there, or if security is lax;
  - via modem, if the central tabulator[177] that programs the cards is connected to a telephone line; or

  - via modem if the PCOS that reads the cards is connected to a telephone line.

- The attacker programs the memory cards to generate a vote total that switches several votes from the Democratic-Republicans to the Federalists (or from Jefferson to Adams).

- He further instructs the memory card to generate the false total only if 400 ballots have run through the scanner in a single 24-hour period (unlike DREs, PCOS scanners can scan hundreds or thousands of votes in a single day). This should help it avoid detection during L&A Testing.

- The attacker does not have to worry about ITA inspection and testing, or Acceptance testing, because the memory cards are not subject to ITA inspection or testing, and are created after acceptance testing is complete.

- At the close of the polls, when election officials and/or poll workers ask the PCOS scanner to generate its vote total report, the false report would be generated.

As with Trojan Horse Attacks and other Software Attack Programs used against DREs, the attackers could target a relatively small number of machines and still change the outcome of our statewide race.

We have assumed that the State of Pennasota has purchased one PCOS machine for each precinct.[178] This would mean that in its three largest counties, there would be a total of 1,669 PCOS machines, with approximately 693 voters per machine. In the entire state, there would be 4,820 machines, with approximately 718 voters per machines.[179]

Again, presuming that our attacker wants to switch 51,891 votes from Tom Jefferson to Johnny Adams, she could target fewer than half of the machines in the three largest counties, switching about 7% of the votes for governor on each machine.[180] On the other hand, if the attacker chose to target all PCOS scanners in the state, it would be necessary to switch only about 8 votes per machine (or slightly more than 1% of all votes cast on each machine).[181]

As with the Software Attacks against DREs previously discussed, if the Software Attack Program functioned as intended (and presuming there was no recount, Parallel Testing or audit), there would be no way for election officials to know that the electronic records were tampered with.

This attack would require a minimum of one to three people: one if the central tabulators in several counties are connected to a telephone line (in which case, an attack could hack into the central tabulators and insert the attack program into the memory cards via the central tabulator), and three if the state made sure that there was no way to contact the central tabulators or PCOS machines via modem or wireless communication (in which case, three individuals would have to gain access to the county offices in the three largest counties and program or reprogram the memory cards before they were sent to the polling places).

■■■ EFFECT OF BASIC SET OF COUNTERMEASURES

Our analysis of the three sets of countermeasures is substantially similar to our analysis in the DRE w/ VVPT section.

This attack is unlikely to be caught by the Basic Set of Countermeasures. Memory cards are not subject to ITA or Acceptance testing. If the attacker is clever, she should be able to ensure that Logic and Accuracy Testing does not catch this attack either. The memory cards are inserted in the normal course of election practice; physical security around the machines and ballots would not prevent successful execution of the attack.

■■■ EFFECT OF REGIMEN FOR PARALLEL TESTING PLUS BASIC SET OF COUNTERMEASURES

We are unaware of any jurisdiction that performs Parallel Testing on PCOS systems. Nevertheless, we believe that Parallel Testing would probably catch this attack. Unlike Trojan Horses and other Software Attack Programs previously discussed, the attack would probably not allow the PCOS to know whether it was being parallel tested.[182]

However, our concerns regarding the ability of other types of Software Attack Programs to circumvent Parallel Testing (*i.e.,* the insertion of a Trojan Horse into firmware, vendor software, COTS software, software patches and updates) apply to PCOS for the same reasons already detailed in our discussion of attacks against DREs. Specifically, we believe that, under the right circumstances and with enough knowledge and time, it would be possible to devise a Software Attack Program against PCOS systems that would allow the scanners to trigger or deactivate based upon the program's ability to detect whether the scanner is being tested.

Thus, if the attacker knew that Parallel Testing was performed on PCOS machines in Pennasota, she could insert a Trojan Horse that would recognize if the machine was being Parallel Tested. *This would require involving between one and three additional people in the attack:* specifically the attack would need to involve people who could gain enough knowledge about the Parallel Testing regime (*i.e.,* the Parallel Testing script writer, a consultant who worked on creating the Parallel Testing procedures) to provide information to subvert it.

EFFECT OF REGIMEN FOR AUTOMATIC ROUTINE AUDIT PLUS BASIC SET OF COUNTERMEASURES

All of our findings regarding the Regimen for Automatic Routine Audit in the DRE w/ VVPT section apply to the Automatic Routine Audit as a countermeasure against the least difficult attack against PCOS. If the Regiment for Automatic Routine Audit is fully implemented (including the use of transparent randomness in selecting auditors and polling places for audit, as well as instituting proper chain of custody and paper security practices), *the Regimen for Automatic Routine Audit Plus Basic Set of Countermeasures should make the least difficult attack against PCOS more difficult by several hundred participants.*

However, at least two of the attacks in our attack catalog point us to unique issues associated with PCOS and the Regiment for Automatic Routine Audit countermeasures.

PCOS Attack Number 42: Trojan Horse Disables Overvote Protections

One of the benefits of PCOS machines over Central Count Optical Scanners (which are very often used in tallying absentee ballots) is that it has an "over/undervote protection." The attack discussed below is a variant of the Trojan Horse attacks already discussed[183] with one important exception: instead of changing votes or the vote total tally, it merely disables the over/undervote protection.

The over/undervote protection on PCOS scanners works as follows: when a voter fills out his ballot, but accidentally skips a race (or accidentally fills in two candidates for the same race), the scanner would refuse to record the vote and send it back to the voter for examination. The voter than has the opportunity to review the ballot and correct it before resubmitting.

Central Count Optical Scanners have been shown to lose as many as three times as many votes as PCOS.[184] The lack of over/undervote protection on Central Count Optical Scanners may be the reason for this difference. In counties with over 30% African American voters, the lost or "residual" vote rate has been shown to be as high as 4.1%.[185]

Our attacker in Pennasota would probably not be able to swing the gubernatorial race from Jefferson to Adams merely by inserting an Attack Program that would turn off the over/undervote protection on PCOS scanners. Even if we assume that the result of turning off the protection were a loss of 4% of the votes on every scanner, and that all of those votes would have gone to Tom Jefferson, this would only result in the loss of about 20,000 votes. This would still leave Jefferson (who won by over 80,000) with a comfortable (though slimmer) margin of victory.

Nevertheless, this attack could cause the loss of thousands of votes, disproportionately affecting poor and minority voters. Neither the Basic Set nor Automatic Routine Audit Plus Basic Set countermeasures (without some sort of statistical analysis of over/undervotes) would counter this attack.

There are at least two possible ways to catch this attack:

- Through Parallel Testing (assuming that the Attack Program has not also figured out a way to shut off when it is being tested); and

- By counting over/undervotes in the audit of the voter-verified paper record to determine whether there is a disproportionate number of such lost votes *(this again points to the importance of statistical analysis and investigation in conjunction with the audit of the voter-verified paper record – by looking for an unusual number of over- and undervotes, the state could spot this kind of attack)*.

PCOS Attack Number 49: Attack on Scanner Configuration Causes Misrecording of Votes

Advocates for PCOS systems point out that the paper record is created by the voter, rather than a machine; the purported benefit of voter-created paper records is that they cannot be corrupted by the machine (as in DRE w/ VVPT Attack Number 6, where the machine creates an incorrect paper record).

The flip side of this benefit is that, in filling out their ballots, people can make mistakes: they might circle the oval instead of filling it in; they might fill in only half the oval; they might fill the oval in with a pencil that the machine cannot recognize. If our attackers configured our machines so that they tended to read partially filled ovals for Johnny Adams, but not Tom Jefferson, Johnny Adams could benefit with many additional votes. Given our analysis of PCOS Attack Number 8, we are skeptical that this attack would be sufficient to turn our imagined election from Jefferson to Adams (though without more investigation, we are unable to come to a certain conclusion). Nevertheless, we are confident that if PCOS Attack Number 49 were accomplished via an Attack Program that reached every PCOS scanner, it probably could affect thousands of votes.

This attack highlights a problem that is unique to the PCOS system. In conducting an audit of the voter-verified paper record or recount, what should be counted as a vote? If the test is merely what the machine reads as a vote, Attack Number 49 would succeed without further investigation.

Again, some statistical analysis done in conjunction with the Automatic Routine Audit (perhaps allowing the Secretary of State's office to review ballot images to look for discrepancies in how votes are counted by the scanners) should allow a jurisdiction to catch this attack.

CONCLUSIONS

Conclusions from Representative Least Difficult Attacks

With the Basic Set of Countermeasures in place, a minimum of 1 to 3 informed participants would be needed to successfully execute PCOS Attack #41 (Software Attack on Inserted Memory Cards) and change the result of the Pennasota Governor's race.

With the Basic Set of Countermeasures Plus the Parallel Testing Regimen in place, PCOS Attack #41 becomes more difficult. The attacker will need at least 3 to 7 informed participants to successfully execute this attack and change the result of the Pennasota Governor's race.

PCOS Attack #41 would be substantially more difficult to successfully execute against the Basic Set of Countermeasures Plus the Automatic Routine Audit Regimen than it would be against the Basic Set of Countermeasures or the Basic Set of Countermeasures plus the Parallel Testing Regimen. The attacker will need at least 386 informed participants to successfully execute PCOS Attack #41 and change the result of the Pennasota Governor's race.

Conclusions about PCOS

- As with DREs, local jurisdictions that take more control of running their own elections (by performing their own programming, creating their own ballot definition files, *etc.*), are going to make successful attacks against statewide elections more difficult.

- The value of paper ballots without the Automatic Routine Audits is highly questionable.

- If voters are well informed as to how to properly fill out PCOS ballots, many attacks against PCOS systems will become more difficult.

Conclusions about the Regimen for Automatic Routine Audit Countermeasure

- Statistical examination of anomalies in ballot images and vote totals, such as higher than expected over and undervotes, can help detect fraud. Currently, none of the states that conduct Automatic Routine Audits examine paper records for statistical anomalies.

- Automatic Routine Audits conducted soon after the close of polls are less vulnerable to attack, because there is less time to tamper with the paper records.

- Solid chain of custody practices and physical security of paper records prior to the Automatic Routine Audit are crucial to creating an effective auditing regimen. The practices discussed on pages 87 and 88, *infra*, should assist jurisdictions in creating an effective auditing regimen.

- The auditing process will be much less vulnerable to attack if machines and auditors are selected and assigned in a publicly transparent manner.

Conclusions about Taking Action When Anomalies Are Found in the Automatic Routine Audit

As is the case for DREs w/ VVPT, an Automatic Routine Audit of PCOS ballots offers questionable security benefit unless effective procedures to investigate discrepancies (including taking remedial action, where necessary) are implemented and adhered to. Detection of possible fraud without an effective response will not thwart an attack on voting systems. For further discussion of this topic, *see supra* pp. 74–75.

PREVENTION OF WIRELESS COMMUNICATION: A POWERFUL COUNTERMEASURE FOR ALL THREE SYSTEMS

Against all three systems, attackers could use wireless components to subvert *all* testing.

As already discussed in some detail (*see supra* at pp. 46, 48, 55, 56), our analysis shows that machines with wireless components are particularly vulnerable to Trojan Horse and other attacks. We conclude that this danger applies to all three systems we have examined. Only two states, New York and Minnesota, ban wireless components on all machines.[186] California's ban on wireless components appears to apply to DREs only.[187]

Unfortunately, banning *use* of wireless components on voting systems without banning the wireless components themselves (as is done in several states) still poses serious security risks. First, a Software Attack Program could be designed to re-activate any disabling of the wireless component. In such circumstances, the voting machine might indicate that the wireless component was off, when it actually could receive signals. Second, pollworkers or anyone else with access to the voting machine could turn on the wireless component when it was supposed to be turned off. Under either scenario, our attacker could use a wireless-enabled PDA or other device to send remote signals to the wireless component and install her attack.

Vendors continue to manufacture and sell machines with wireless components.[188] Among the many types of attacks made possible by wireless components are attacks that exploit an unplanned vulnerability in the software or hardware to get a Trojan Horse into the machine. For this type of attack, an attacker would not need to insert a Trojan Horse in advance of Election Day. Instead, if she was aware of a vulnerability in the voting system's software or firmware, she could simply show up at the polling station and beam her Trojan Horse into the machine using a wireless-enabled PDA.

Thus, virtually any member of the public with some knowledge of software and a PDA could perform this attack. This is particularly troubling when one considers that most voting machines run on COTS software and/or operating systems; the vulnerabilities of such software and systems are frequently well known.[189]

Against all three systems, attackers could use wireless components to subvert *all* testing. Specifically, an attack program could be written to remain dormant until it received specific commands via a wireless communication. This would allow attackers to wait until a machine was being used to record votes on Election Day before turning the software attack on.

Attackers could also use wireless communications to gain fine-grained control over an attack program already inserted into a particular set of machines (*i.e.,* switch three votes in the second race on the third machine), or obtain information as to how individuals had voted by communicating with a machine while it was being used.

Finally, wireless networking presents additional security vulnerabilities for jurisdictions using DREs w/ VVPT and PCOS. A major logistical problem for an attacker changing both electronic and paper records is how to get the new paper records printed in time to substitute them for the old record in transit. With wireless networking, the DRE or PCOS can transmit specific information out to the attacker about what should appear on those printed records. In short, permitting wireless components on VVPT or PCOS machines makes the attacker's job much simpler in practice.

SECURITY RECOMMENDATIONS

There is a substantial likelihood that the election procedures and countermeasures currently in place in the vast majority of states would not detect a cleverly designed Software Attack Program. The regimens for Parallel Testing and Automatic Routine Audits proposed in the Security Report are important tools for defending voting systems from many types of attack, including Software Attack Programs. For the reasons discussed, *infra* at pp. 6–7, we also believe that these measures would reduce the likelihood that votes would be lost as a result of human error.

There is a substantial likelihood that the election procedures and countermeasures currently in place in the vast majority of states would not detect a cleverly designed Software Attack Program.

Most jurisdictions have not implemented these security measures. Of the 26 states that require a voter-verified paper record, only 12 states require automatic audits of those records after every election, and only two of these states – California and Washington – conduct Parallel Testing. [190] Moreover, even those states that have implemented these countermeasures have not developed the best practices and protocols that are necessary to ensure their effectiveness in preventing or revealing attacks or failures in the voting systems.

Recommendation #1:
Conduct Automatic Routine Audit of Paper Records.

Advocates for voter-verified paper records have been extremely successful in state legislatures across the country. Currently, 26 states require their voting systems to produce a voter-verified record, but 14 of these states do not require Automatic Routine Audits. [191] The Task Force has concluded that an independent voter-verified paper trail without an Automatic Routine Audit is of questionable security value.[192]

By contrast, a voter-verified paper record accompanied by a solid Automatic Routine Audit can go a long way toward making the least difficult attacks much more difficult. Specifically, the measures recommended below should force an attacker to involve hundreds of more informed participants in her attack.

- A small percentage of all voting machines and their voter-verified paper records should be audited.
- Machines to be audited should be selected in a random and transparent way.
- The assignment of auditors to voting machines should occur immediately before the audits. The audits should take place by 9 a.m., the day after polls close.
- The audit should include a tally of spoiled ballots (in the case of VVPT cancellations), overvotes, and undervotes.

For paperless DRE voting machines, Parallel Testing is probably the best way to detect most software-based attacks.

- A statistical examination of anomalies, such as higher than expected cancellations or over- and undervotes, should be conducted.

- Solid practices with respect to chain of custody and physical security of paper records prior to the Automatic Routine Audit should be followed.

Recommendation #2: Conduct Parallel Testing.

It is not possible to conduct an audit of paper records of DREs without VVPT, because no voter-verified paper record exists on such machines. This means that jurisdictions that use DREs without VVPT do not have access to an important and powerful countermeasure.

For paperless DRE voting machines, Parallel Testing is probably the best way to detect most software-based attacks, as well as subtle software bugs that may not be discovered during inspection and other testing. For DREs w/ VVPT and ballot-marking devices, Parallel Testing provides the opportunity to discover a specific kind of attack (for instance, printing the wrong choice on the voter-verified paper record) that may not be detected by simply reviewing the paper record after the election is over. However, even under the best of circumstances, Parallel Testing is an imperfect security measure. The testing creates an "arms race" between the testers and the attacker, but the race is one in which the testers can never be certain that they have prevailed.

We have concluded that the following steps will lead to more effective Parallel Testing:

- The precise techniques used for Parallel Testing (*e.g.*, exactly how and when the machine is activated, how activation codes/smart cards/*etc.* are produced to allow voting, *etc.*) should not be fully determined or revealed until right before the election. Details of how Parallel Testing is done should change from election to election.

- At least two of each type of DRE (meaning both vendor and model) should be selected for Parallel Testing.

- At least two DREs from each of the three largest counties should be parallel tested.

- Localities should be notified as late as possible that machines from their precincts will be selected for Parallel Testing.

- Wireless channels for voting machines should be closed off, to ensure they cannot receive commands.

- Voting machines should never be connected to one another during voting.[193]

- Voting machines should be completely isolated during the election, and print out or otherwise display their totals *before* being connected to any central server to send in its tallies.

- Parallel Testing scripts should include details, such as how quickly or slowly to vote, when to make "errors," and perhaps even when to cast each vote.

- Parallel Testing should be videotaped to ensure that a contradiction between paper and electronic records when Parallel Testing is complete is not the result of tester error.

While a few local jurisdictions have taken it upon themselves to conduct limited Parallel Testing, we are aware of only three states, California, Maryland and Washington, that have regularly performed Parallel Testing on a statewide basis. It is worth noting that two of these states, California and Washington, employ Automatic Routine Audits *and* Parallel Testing as statewide countermeasures against potential attack.

Recommendation # 3:
Ban Wireless Components on All Voting Machines.

Machines with wireless components are particularly vulnerable to attack.

Our analysis shows that machines with wireless components are particularly vulnerable to attack. We conclude that this vulnerability applies to all three voting systems. Only two states, New York and Minnesota, ban wireless components on all machines.[194] California also bans wireless components, but only for DRE machines. Wireless components should not be permitted on any voting machine.

Recommendation # 4:
Mandate Transparent and Random Selection Procedures.

The development of transparently random selection procedures for all auditing procedures is key to audit effectiveness. This includes the selection of machines to be parallel tested or audited, as well as the assignment of auditors themselves. The use of a transparent and random selection process allows the public to know that the auditing method was fair and substantially likely to catch fraud or mistakes in the vote totals. In our interviews with election officials we found that, all too often, the process for picking machines and auditors was neither transparent nor random.

In a transparent random selection process:

- The whole process is publicly observable or videotaped.

- The random selection is to be publicly verifiable, *i.e.*, anyone observing is able to verify that the sample was chosen randomly (or at least that the number selected is not under the control of any small number of people).

- The process is simple and practical within the context of current election practice so as to avoid imposing unnecessary burden on election officials.

Recommendation # 5: Ensure Local Control of Election Administration.

Where a single entity, such as a vendor or state or national consultant, runs elections or performs key tasks (such as producing ballot definition files) for multiple jurisdictions, attacks against statewide elections become easier. Unnecessary centralized control provides many opportunities to implement attacks at multiple locations.

Recommendation # 6: Implement Effective Procedures for Addressing Evidence of Fraud or Error.

Both Automatic Routine Audits and Parallel Testing are of questionable security value without effective procedures for action where evidence of machine malfunction and/or fraud is uncovered. Detection of fraud without an appropriate response will not prevent attacks from succeeding. In the Brennan Center's extensive review of state election laws and practices, and in its interviews with election officials for the Threat Analysis, we did not find any jurisdiction with publicly detailed, adequate, and practical procedures for dealing with evidence of fraud or error discovered during an audit, recount or Parallel Testing.

The following are examples of procedures that would allow jurisdictions to respond effectively to detection of bugs or Software Attack Programs in Parallel Testing:

- Impound and conduct a transparent forensic examination of all machines showing unexplained discrepancies during Parallel Testing.

- Where evidence of a software bug or attack program is subsequently found (or no credible explanation for the discrepancy is discovered), conduct a forensic examination of all DREs in the state used during the election.[195]

- Identify the machines that show evidence of tampering or a software flaw that could have affected the electronic tally of votes.

- Review the reported margin of victory in each potentially affected race.

- Based upon the (1) margin of victory, (2) number of machines affected, and (3) nature and scope of the tampering or flaw, determine whether there is a substantial likelihood that the tampering or flaw changed the outcome of a particular race.

- Where there is a substantial likelihood that tampering changed the outcome of a particular race, hold a new election for the office.

The following is an illustrative set of procedures that would allow jurisdictions to respond effectively to discrepancies between paper and electronic records during an Automatic Routine Audit:

- Conduct a transparent investigation of all machines where the paper and electronic records do not match to determine whether there is any evidence that tampering with the paper records has occurred.

- To the extent that there is no record that the paper records have been tampered with, certify the paper records.

- If there is evidence that the paper records have been tampered with, give a presumption of authority to the electronic records.

- After giving a presumption of authority to the electronic records, conduct a forensic investigation on all machines where the paper and electronic records do not match, to determine whether there has been any tampering with the electronic records.

- If tampering with the electronic records can be ruled out, certify the electronic records.[196]

- Where there is evidence that both sets of records have been tampered with, conduct a full recount to determine whether and to what extent paper and electronic records cannot be reconciled.

- At the conclusion of the full recount, determine the total number of machines that report different electronic and paper records.

- After quantifying the number of machines that have been tampered with, determine the margin of victory in each potentially affected race.

- Based upon (1) the margin of victory, (2) the number of machines affected, and (3) the nature and scope of the tampering, determine whether there is a substantial likelihood that tampering changed the outcome of a particular race.

- In the event that a determination is made that there is a substantial likelihood that tampering changed the outcome of a particular race, hold a new election for the office.

Election officials and voting systems experts should be looking at ways to ensure that voters understand their role in creating a more secure voting system.

DIRECTIONS FOR THE FUTURE

We are hopeful that this report will spur further orderly and empirical analyses of threats to voting systems, for the purpose of assessing new voting systems, as well as proposed security procedures and countermeasures. Some of our suggestions for further study are detailed below.

WITNESS AND CRYPTOGRAPHIC SYSTEMS

This report was necessarily limited to analyzing systems currently in use. Further security analyses must be performed on witness and cryptographic voting systems, which provide some hope of offering election officials additional choices for independently verifiable voting systems in the future.

For a detailed discussion of these systems and their potential, *see* the website of the Electronic Privacy Information Center at http://www.epic.org/privacy/voting/eac_foia/v1ad.doc. Also *see* the website of the Society for Industrial and Applied Mathematics at http://www.siam.org/siamnews/04-04/voting.pdf.

INFORMING VOTERS OF THEIR ROLE IN MAKING SYSTEMS MORE SECURE

This report makes clear that informed voters are an important defense against potential attacks. The larger the number of voters who check their VVPT before casting their vote, the less likely that an Automatic Routine Audit would be unable to catch a Trojan Horse attack. Similarly, the more voters who fill out their PCOS ballots correctly, the less likely that a Trojan Horse attack on the over/undervote protection or scanner calibration will affect the number of recorded votes.

Election officials and voting systems experts should be looking at ways to ensure that voters understand their role in creating a more secure voting system.

ADDITIONAL STATISTICAL TECHNICAL TECHNIQUES TO DETECT FRAUD

This study has pointed to at least two areas where statistical techniques in the Automatic Routine Audit could be used to catch fraud: (1) where there is an unusually high number of cancellations on the VVPT and (2) where there is an unusually high number of under/overvotes on PCOS ballots. We encourage statisticians and political scientists to find additional statistical techniques to detect fraud.

LOOKING FOR BETTER PARALLEL TESTING TECHNIQUES

We conclude that Parallel Testing can be a useful countermeasure that should make voting systems more secure, particularly in jurisdictions where voting systems do not have voter-verified paper records. We have made a number of observations concerning solid Parallel Testing practices. We believe that additional studies should be done to attempt to make Parallel Testing practices even stronger. Parallel Testing creates an "arms race" of sorts between the testers and the attacker – where the testers can never be certain that they have prevailed.

LOOKING AT OTHER ATTACK GOALS

This report took on the simplifying assumption that the attacker's objective was to change the outcome of a statewide race. But attackers could have other goals: to attack voter privacy, disrupt an election, or discredit the electoral process. All of these are serious threats that we should guard against. Methodical threat analyses of these attack objectives would also be useful and employing the same approach used here might well provide critical insight.

LOOKING AT OTHER RACES

The method and analysis of this study can be applied to any race, real or hypothetical, local or statewide.[197] We encourage security analysts, public officials and interested citizens to use the information and methods in this document to address their specific security concerns.

GLOSSARY[198]

Automatic Routine Audit. Automatic Routine Audits are used in twelve states to test the accuracy of electronic voting machines. They generally require between 1 and 10% of all precinct voting machines to be audited.[199] The Task Force findings regarding Automatic Routine Audit regimens can be found in this report at pages 76–77, and 87–88.

Cryptic or Secret Knock. Where a Trojan Horse or other Software Attack Program has been inserted into a machine, a Cryptic Knock is an action taken by a user of the machine that will trigger (or silence) the attack behavior. The Cryptic Knock could come in many forms, depending upon the attack program: voting for a write-in candidate, tapping a specific spot on the machine's screen, a communication via wireless network, *etc.*

Configuration Files. Voting systems are generally designed to be used across many jurisdictions with very different needs, regulations and laws. In addition to the ballot definition information in a voting terminal on Election Day, there are a wide range of settings that must be configured correctly in order to be have the terminal perform correctly. For instance, machines must be configured to tell the system how to behave when a voter leaves with a ballot not completed, and the election officials indicate to the machine that the voter fled without casting his ballot. In some jurisdictions, the machine should cast the ballot while in others, it should void the ballot. These settings can be thought of as residing in configuration files, although they may actually be stored in the Windows Registry, in a database or elsewhere.

Driver. In general, a driver is a program designed to interface a particular piece of hardware to an operating system or other software. Computers systems are designed with drivers so that many programs such as MS Word, QuickBooks, and Firefox web browser, for example, could interface to lots of devices such as printers, monitors, plotters, and barcode readers without having to have each one of these programs depend on the details of each device. With regard to voting technology, drivers are likely to be present to interface with audio devices for accessibility, the screen, the touch-screen hardware, a printer for printing totals, and other information as well and for interfacing with the battery backup unit.

Event and Audit Logs. In general, computer systems are programmed to record all activities that occur – including when they are started up, when they are shut down, *etc.* A voting terminal could be programmed to remember when it was started, shutdown, when it printed its zero tape, and the like. Such records are Event Logs or Audit Logs. These records could be helpful during a forensic analysis of voting systems after a suspected attack.

Independent Testing Authority. Starting with the 1990 FEC/NASED standards, independent testing authorities ("ITAs") have tested voting systems, certifying that these systems meet the letter of the "voluntary" standards set by the federal government and required, by law, in most states. Several states, such as Florida, that impose additional standards contract with the same labs to test to these stronger standards.[200]

Logic and Accuracy Testing (or "L&A" Testing"). This is the testing of the tabulator setups of a new election definition to ensure that the content correctly reflects the election being held (*i.e.,* contests, candidates, number to be elected, ballot formats, *etc.*) and that all voting positions can be voted for the maximum number of eligible candidates and that results are accurately tabulated and reported.[201] Logic and Accuracy Testing should not be confused with Parallel Testing. Logic and Accuracy Testing is generally done prior to the polls opening; it is not intended to mimic the behavior of actual voters and generally lasts only a few minutes. Most machines have a "Logic and Accuracy" setting, so that the machine "knows" it is being tested.

Parallel Testing. Parallel testing, also known as election-day testing, involves selecting voting machines at random and testing them as realistically as possible during the period that votes are being cast. The Task Force findings regarding parallel testing regimens can be found in this report at pages 53–61, and 88–89.

Software Attack Program. Any destructive program, including Trojan Horses, viruses, or other code that is used to overtake voting systems for the purpose of altering election results.

Trojan Horse. A destructive program that masquerades as a benign program. Unlike viruses, Trojan Horses do not replicate themselves.[202]

APPENDIX A

ALTERNATIVE THREAT ANALYSIS MODELS CONSIDERED

Measuring the complexity of the trusted computing base.

Before adopting the threat model discussed in this report, The Task Force considered other potential methods of analysis, including measuring the complexity of the trusted computing base. In computer security terminology, the *trusted computing base* (the "TCB") is the "totality of protection mechanisms within a computing system including hardware, firmware and software, the combination of which is responsible for enforcing a security policy."[203]

For many Task Force members, evaluating the complexity of the TCB was an attractive method for evaluating the relative security of different voting systems. In essence, this methodology would look at how "complicated" the trusted computing base of each system was, by reviewing code and other technological complexities. The more complex the TCB, the more likely that it could be attacked without notice.

We quickly realized that this was not a satisfactory way to analyze the relative security of systems. If we only looked at the complexity of the voting system TCB in analyzing its vulnerabilities, we would come to some very strange conclusions and ignore some important historical lessons about election fraud. For instance, under this system of analysis, the hand counting of ballots would carry no risk (there would be no TCB under this system). In fact, as election officials know all too well, pure paper elections have repeatedly shown themselves to be vulnerable to election fraud.[204]

While it may be wise to minimize the total amount of technology we "trust" in elections, as a method for assessing the strength of a voting system and identifying potential weaknesses, it does not appear to provide a useful means of analysis.

Counting points of vulnerability.

A related methodology would be to look at the points of vulnerability within a system. At first blush, this also appeared to be an attractive method for a security analysis. Obviously, we would like to minimize the ways that an attacker might compromise an election. It is easier to guard one door than a thousand.

As a practical matter, however, it did not appear to be a very good way to prioritize threats, or identify vulnerabilities that election officials should be most worried about. Obviously a system with three highly vulnerable points that are impossible to protect is not preferable to a system with four small points of vulnerability that are easy to protect.

Examining Adherence to NIST Risk Assessment Controls.

This model would compare voting systems with guidelines established in NIST Special Publication 800-30, Risk Management Guide for Information Technology Systems. Special Publication 800-30 provides a generic methodology for examining, assessing, and mitigating risk. However, it does not specifically address threats and vulnerabilities unique to the voting environment. For this reason, the Task Force rejected it as a basis for establishing a voting systems threat analysis model.

APPENDIX B
VOTING MACHINE DEFINITIONS

Direct Recording Electronic Voting Machine

A Direct Recording Electronic ("DRE") voting machine directly records the voter's selections in each race or contest. It does so via a ballot that appears on a display screen. Typical DRE machines have flat panel display screens with touch-screen input, although other display technologies have been used (this includes paper and push button displays). The defining characteristic of these machines is that votes are captured and stored electronically.

Software is updated in DRE systems via various methods, specific to each voting system. In general, software updating involves someone (usually a technician or election official representative) installing new software over older software using whatever medium the DRE uses to transport votes (sometimes, it is done using laptop computers, using special software provided from vendors).

Examples of DRE systems include: Hart InterCivic's eSlate, Sequoia's AVC Edge, ES&S's iVotronic, Diebold AccuVote-TS and AccuVote TSx, AVS WinVote and Unilect Patriot.

Direct Recording Electronic Voting Machine with Verified Voter Paper Trail

A Direct Recording Electronic Voting Machine with Voter-Verified Paper Trail ("DRE w/ VVPT") is a DRE that captures a voter's choice both (1) internally in purely electronic form, and (2) contemporaneously on paper, as a voter-verified record. A DRE w/ VVPT allows the voter to view and confirm the accuracy of the paper record.

Examples of DRE w/ VVPT include: AccuPoll, AvanteVote-Tracker EVC-308SPR, Sequoia VeriVote with Printer attachment, TruVote and Diebold Accuview with VVPT Printer attachment.

Precinct Count Optical Scan

Precinct Count Optical Scan ("PCOS") is a voting system that allows voters to mark paper ballots, typically with pencils or pens. Voters then carry their ballots (sleeved or otherwise protected so that others cannot see their choices) by hand to a scanner. At the scanner, they un-sleeve the ballot and insert it into the scanner, which optically records the vote.

Examples of PCOS include: Avante Optical Code Tracker, ES&S Model 100, Sequoia or ES&S Opteck II-P Eagle, Diebold AccuVote-OS.

APPENDIX C

ALTERNATIVE SECURITY METRICS CONSIDERED

Dollars Spent

The decision to use the number of informed participants as the metric for attack level difficulty came after considering several other potential metrics. One of the first metrics we considered was the dollar cost of attacks. This metric makes sense when looking at attacks that seek financial gain – for instance, misappropriating corporate funds. It is not rational to spend $100,000 on the misappropriation of corporate funds if the total value of those funds is $90,000. Ultimately, we rejected this metric as the basis for our analysis because the dollar cost of the attacks we considered were dwarfed by both (1) current federal and state budgets, and (2) the amounts currently spent legally in state and federal political campaigns.

Time of Attack

The relative security of safes and other safety measures are often rated in terms of "time to defeat." This was rejected as metric of difficulty because it did not seem relevant to voting systems. Attackers breaking into a house are concerned with the amount of time it might take to complete their robbery because the homeowners or police might show up. With regard to election fraud, many attackers may be willing to start months or years before an election if they believe they can control the outcome. As discussed *supra* at pp. 35–48, attackers may be confident that they can circumvent the independent testing authorities and other measures meant to identify attacks, so that the amount of time an attack takes becomes less relevant.

APPENDIX D

BRENNAN CENTER SECURITY SURVEY

1. Do you request that your responses remain anonymous?

   ❑ yes ❑ not necessary

2. What type of machine(s) did you use in the last election (please indicate make, model and type)? And do you expect to use different machines within the next two years (if yes, indicate which new machines you expect to use)?

3. Does your jurisdiction provide voters with sample ballots before Election Day?

4. What security measures does your jurisdiction take related to the storage of voting machines?

   a. Are machines stored in a secure location? If so, in what type of location are they stored and how are they made secure?

   b. Are there tamper evident seals placed on machines? If so, when are they placed around machines? When are they taken off?

   c. Is inventory of machines taken at any time between elections?

   d. Other security measures during storage? If so, please detail these security measures.

5. What security measures does your jurisdiction take when transporting machines to polling place?

   a. How and by whom are the machines transported?

   b. How long between transportation and use on Election Day?

   c. Other security measures during transportation? If so, please detail these security measures.

6. What, if any, testing is done to ensure that the machines are properly recording and tallying votes ("logic and accuracy testing") of machines prior to or on Election Day? If testing is done, please detail who does testing and how it is done.

7. What, if any, security measures do you take on Election Day immediately prior to opening polls?

    a. Inventory of machines, parts (please indicate which parts)?

    b. Check clock on machines?

    c. Check ballots to ensure correct precinct?

    d. Record number of ballots?

    e. Print and sign zero tape?

    f. Other security measures immediately prior to opening polls? If so, please detail these security measures.

8. What, if any, security measures do you take during the period in which polls are open?

    a Entry and exit of each voter to/from polling place recorded in poll books?

    b. If you use DRE with paper trail, is each voter encouraged to verify the accuracy of the paper receipt? If so, how?

    c. If machine is OpScan, is anything done to ensure that overvote protection is not turned off manually? If so, what is done?

    d. If machine is OpScan, is there a stated/written policy for how poll workers should deal with a ballot that is rejected by the machine because of an overvote? If so, what is that policy?

    e. If you use DRE with verified paper trail or OpScans, how is ballot/paper stored after votes have been cast on Election Day?

    f. If there are ballots or machine produced paper, what is done with "spoiled" ballots/paper?

    g. Other security measures taken on Election Day? If so, please detail these security measures.

9. What if any security measures are taken at close of Election Day?

    a. If you have cartridges with ballot images, are these collected to ensure that number of cartridges matches number of machines?

b. Are numbers of blank and spoiled ballots determined?

c. Do poll workers sign ballot tapes? If so, when?

d. How are vote tallies in polling place reported to central office (*e.g.*, phone, modem, other method)?

e. What measures are taken to ensure that polling place vote tallies are accurately recorded at central office?

f. What is done with (i) machine cartridges, (ii) machine tapes and (iii) poll books at close of election? Are these placed in a secure location? If so, how do you make placement secure (please answer separately for each)?

g. What measures are taken to ensure that valid provisional ballots are accurately counted and secured for potential recounts?

h. If you use OpScan or DRE with a verified paper trail, what is done with these ballots/papers at close of Election Day?

i. Is there any public posting of polling place tallies by individual polling places (other than report to central office)? If so, where is this posting made?

j. What is done with machines at close of the polls, after votes have been counted?

k. Other security measures after close of Election Day? If so, please detail these security measures.

10. The Brennan Center is currently conducting research about voting machines in a variety of areas, including voting machine security. We would very much like to have the insights of election officials, who understand the practical concerns of running an election and ensuring that it is conducted as securely as possible.

We may want to follow up by telephone or e-mail to ask about your responses. Would you have any objection to this?

County, State: ______________________________

Name/Title:______________________________

Phone/e-mail: ______________________________

Best time to follow up:______________________________

APPENDIX E

VOTING MACHINE TESTING

An Overview of Voting Machine Testing[205]

Voting systems are subjected to many tests over their lifetimes, beginning with testing done by the manufacturer during development and ending on Election Day. These tests are summarized below, along with a brief description of the strengths and weaknesses of each test.

- Internal testing at the vendor
- Independent Testing Authority certification
- State qualification tests
- Tests conducted during contract negotiation
- Acceptance testing as delivered
- Pre-election (Logic and Accuracy) testing
- Testing as the polls are opened
- Parallel Testing during an election
- Post-election testing

Internal Testing at the Vendor

All responsible product developers intensively test their products prior to allowing any outsiders to use or test them. The most responsible software development methodologies ask the system developers to develop suites of tests for each software component even before that component is developed. The greatest weakness of these tests is that they are developed by the system developers themselves, so they rarely contain surprises.

Independent Testing Authority Certification

Starting with the 1990 FEC/NASED standards, independent testing authorities (ITAs) have tested voting systems, certifying that these systems meet the letter of the "voluntary" standards set by the federal government and required, by law, in most states. Several states, such as Florida, that impose additional standards contract with the same labs to test to these stronger standards.

The ITA process has two primary weaknesses: First, the standards contain many specifics that are easy to test objectively (the software must contain no "naked constants" other than zero and one), and others that are vague or subjective (the software must be well documented). The ITAs are very good at testing to the specific objective requirements, but where subjective judgment or vague requirements are stated, the testing is frequently minimal.

Second, there are many requirements for voting systems that are obvious to observers in retrospect, but that are not explicitly written in the standards (Precinct 216 in Volusia County, Florida reported -16,022 votes for Gore in 2000; prior to this, nobody thought to require that all vote totals be positive). The ITA cannot be expected to anticipate all such omissions from the standards.

Finally, the ITA tests are almost entirely predictable to the developers, as with the vendors internal testing. Barring outright oversights or carelessness on the part of the vendor, and these do occur, and barring the vendors decision to use the ITA process in lieu of an extensive internal testing program, the ITA testing can be almost pro forma. Catching carelessness on the part of vendors and offering a guarantee that minimal standards have been met is sufficiently important that the ITA process should not be dismissed out of hand.

State Qualification Tests

While some states allow any voting system to be offered for sale that has been certified to meet the "voluntary" federal standards, many states impose additional requirements. In these states, vendors must demonstrate that they have met these additional standards before offering their machines for sale in that state. Some states contract out to the ITAs to test to these additional standards, some states have their own testing labs, some states hire consultants, and some states have boards of examiners that determine if state requirements are met.

In general, there is no point in having the state qualification tests duplicate the ITA tests. There is considerable virtue in having state tests that are unpredictable, allowing state examiners to use their judgment and knowledge of the shortcomings of the ITA testing to guide their tests. This is facilitated by state laws that give the board members the right to use their judgment instead of being limited to specific objective criteria. Generally, even when judgment calls are permitted, the board cannot reject a machine arbitrarily, but must show that it violates some provision required by state law.

State qualification testing should ideally include a demonstration that the voting machine can be configured for demonstration elections that exercises all of the distinctive features of that state's election law, for example, straight party voting, ballot rotation, correct handling of multi-seat races, and open or closed primaries, as the case may be. Enough ballots should be voted in these elections to verify that the required features are present.

Tests Conducted During Contract Negotiation

When a jurisdiction puts out a request for bids, it will generally allow the finalists to bring in systems for demonstration and testing. It is noteworthy that federal certification and state qualification tests determine whether a machine meets the legal requirements for sale, but they generally do not address any of the economic issues associated with voting system use, so it is at this time that economic issues must be evaluated.

In addition, the purchasing jurisdiction (usually the county) has an opportunity, at this point, to test the myriad practical features that are not legislated or written into any standards. As of 2004, neither the FEC/NASED standards nor the standards of most states address a broad range of issues related to usability, so it is imperative that local jurisdictions aggressively use the system, particularly in obscure modes of use such as those involving handicapped access (many blind voters have reported serious problems with audio ballots, for example).

It is extremely important at this stage to allow the local staff who will administer the election system to participate in demonstrations of the administrative side of the voting system, configuring machines for mock elections characteristic of the jurisdiction, performing pre-election tests, opening and closing the polls, and canvassing procedures. Generally, neither the voting system standards, or state qualification tests address questions of how easy it is to administer elections on the various competing systems.

Acceptance Testing as Delivered

Each machine delivered by a vendor to the jurisdiction should be tested. Even if the vendor has some kind of quality control guarantees, these are of no value unless the customer detects failures at the time of delivery. At minimum, such tests should include power-on testing, basic user interface tests (do all the buttons work, does the touch-screen sense touches at all extremes of its surface, do the paper-feed mechanisms work, does the uninterruptible power supply work).

By necessity, when hundreds or even thousands of machines are being delivered, these tests must be brief, but they should also include checks on the software versions installed (as self-reported), checks to see that electronic records of the serial numbers match the serial numbers affixed to the outside of the machine, and so on.

It is equally important to perform these acceptance tests when machines are upgraded or repaired as it is to perform them when the machines are delivered new, and the tests are equally important after in-house servicing as they are after machines are returned from the vendor's premises.

Finally, when large numbers of machines are involved, it is reasonable to perform more intensive tests on some of them, tests comparable to the tests that ought to be performed during qualification testing or contract negotiation.

Pre-Election (Logic and Accuracy) Testing

Before each election, every voting machine should be subject to public testing. This is frequently described as Logic and Accuracy Testing or simply L&A Testing, a term that is more appropriate in the realm of punch-card and mark-sense ballot tabulating machines than in the realm of direct recording electronic systems, but the term is used widely, and in many states, it is enshrined in state law.

The laws or administrative rules governing this testing vary considerably from state to state. Generally, central-count paper ballot tabulating machinery can be subject to more extensive tests than voting machines, simply because each county needs only a few such machines. Similarly, precinct-count paper ballot tabulating machinery, with one machine per precinct, can be tested more intensively than voting machines, which may number in the tens per precinct.

An effective test should verify all of the conditions tested in acceptance testing, since some failures may have occurred since the systems arrived in the warehouse. In addition, the tests should verify that the machines are correctly configured for the specifics of this election, with the correct ballot information loaded, including the names of all applicable candidates, races and contests.

The tabulation system should be tested by recording test votes on each machine, verifying that it is possible to vote for each candidate on the ballot and that these votes are tabulated correctly all the way through to the canvass; this can be done, for example, by casting a different number of votes for each candidate or issue position in each race or contest on the ballot.

When multiple machines are configured identically, this part of the test need only be performed in full and manually on one of the identical machines, while on the others, it is reasonable to simplify the testing by verifying that the other machines are indeed configured identically and then using some combination of automated self-test scripts and simplified manual testing.

For mark-sense voting systems, it is important to test the sensor calibration, verifying that the vote detection threshold is appropriately set between a blank spot on the ballot and a dark pencil mark. The calibration should be tested in terms of pencil marks even in jurisdictions that use black markers, because it is inevitable that some voters will use pencils, particularly when markers go dry in voting booths or when ballots are voted by mail. One way to judge the appropriateness of the threshold setting is to see that the system distinguishes between hesitation marks (single dots made by accidentally resting the pencil tip on a voting target) and X or checkmarks, since the former are common accidents not intended as votes, and most state laws allow an X or check to be counted as a vote even though such minimal marks are never recommended.

For touch-screen voting systems, it is important to test the touch-screen calibration, verifying that the machine can sense and track touches over the entire surface of the touch-screen. Typical touch-screen machines have a calibration mode in which they either display targets and ask the tester to touch them with a stylus, or they display a target that follows the point of the stylus as it is slid around the screen.

For voting systems with audio interfaces, this should be checked by casting at least some of the test ballots using this interface. While doing this, the volume control should be adjusted over its full range to verify that it works. Similarly, where multiple display magnifications are supported, at least one test ballot should be voted for each ballot style using each level of magnification. Neither of these tests can be meaningfully performed using automatic self-testing scripts.

The final step of the pre-election test is to clear the voting machinery, setting all vote totals to zero and emptying the physical or electronic ballot boxes, and then sealing the systems prior to their official use for the election.

Ideally, each jurisdiction should design a pre-election test that, between all tested machines, not only casts at least one vote per candidate on each machine, but also produces an overall vote total arranged so that each candidate and each yes-no choice in the entire election receives a different total. Designing the test this way verifies that votes for each candidate are correctly reported as being for that candidate and not switched to other candidates. This will require voting additional test ballots on some of the machines under test.

Pre-election testing should be a public process. This means that the details and rationale of the tests must be disclosed, the testers should make themselves available for questioning prior to and after each testing session, representatives of the parties and campaigns must be invited, and an effort must be made to make space for additional members of the public who may wish to observe. This requires that testing be conducted in facilities that offer both adequate viewing areas and some degree of security.

It is important to assure that the voting machine configuration tested in the pre-election tests is the same configuration used on Election Day. Loading new software or replacing hardware components on a voting machine generally requires the repetition of those parts of the pre-election tests that could possibly depend on the particular hardware or software updates that were made.

Testing as the Polls are Opened

Prior to opening the polls, every voting machine and vote tabulation system should be checked to see that it is still configured for the correct election, including the correct precinct, ballot style, and other applicable details. This is usually determined from a startup report that is displayed or printed when the system is powered up.

In addition, the final step before opening the polls should be to verify that the ballot box (whether physical or virtual) is empty, and that the ballot tabulation system has all zeros. Typically, this is done by printing a zeros report from the machinery. Ideally, this zeros report should be produced by identically the same software and procedures as are used to close the polls, but unfortunately, outside observers without access to the actual software can verify only that the report itself looks like a poll closing report with all vote totals set to zero.

Some elements of the acceptance tests will necessarily be duplicated as the polls are opened, since most computerized voting systems perform some kind of power-on self-test. In some jurisdictions, significant elements of the pre-election test have long been conducted at the polling place.

Observers, both partisan observers and members of the public, must be able to observe all polling place procedures, including the procedures for opening the polls.

Parallel Testing During an Election

Parallel Testing, also known as election-day testing, involves selecting voting machines at random and testing them as realistically as possible during the period that votes are being cast. The fundamental question addressed by such tests arises from the fact that pre-election testing is almost always done using a special test mode in the voting system, and corrupt software could potentially arrange to perform honestly while in test mode while performing dishonestly during a real election.

Parallel Testing is particularly valuable to address some of the security questions that have been raised about Direct Recording Electronic voting machines (for example, touch-screen voting machines), but it is potentially applicable to all electronic vote counting systems.

It is fairly easy to enumerate a long list of conditions that corrupt election software could check in order to distinguish between testing and real elections. It could check the date, for example, misbehaving only on the first Tuesday after the first Monday of November in even numbered years, and it could test the length of time the polls had been open, misbehaving only if the polls were open for at least 6 hours, and it could test the number of ballots cast, misbehaving only if at least 75 were encountered, or it could test the distribution of votes over the candidates, misbehaving only if most of the votes go to a small number of the candidates in the vote-for-one races or only if many voters abstain from most of the races at the tail of the ballot.

Pre-set vote scripts that guarantee at least one vote for each candidate or that guarantee that each candidate receives a different number of votes can be detected by dishonest software. Therefore, Parallel Testing is best done either by using a random distribution of test votes generated from polling data representative of the electorate, or by asking real voters to volunteer to help test the system (perhaps asking each to flip a coin to decide secretly whether they will vote for the candidates they like or for the candidates they think their neighbor likes).

It is important to avoid the possibility of communicating to the system under test any information that could allow the most corrupt possible software to learn that it is being tested. Ideally, this requires that the particular machines to be tested be selected at the last possible moment and then opened for voting at the normal time for opening the polls and closed at the normal time for closing the polls. In addition, mechanical vote entry should not be used, but real people should vote each test ballot, with at least two observers noting either that the test script is followed exactly or noting the choices made. (A video record of the screen might be helpful.)

Parallel Testing at the polling place is a possibility. This maximizes exposure of the testing to public observation and possibly to public participation, an important consideration because the entire purpose of these tests is to build public confidence in the accuracy of the voting system.

However Parallel Testing is conducted, it is important to guard against any possibility of contamination of the official canvass with ballot data from voting machines that were subject to Parallel Testing. By their very nature, these votes are indistinguishable from real votes, excepting the fact that they came from a machine under test. Therefore, physical quarantine of the vote totals from the Parallel Testing is essential. Use of a different color for paper in the printer under test, use of distinctively colored data cartridges, warning streamers attached to cartridges, and similar measures may all be helpful. In addition, if the serial number of the voting machine is tied to its votes through the canvass, a check to make sure that the serial numbers of the machines under Parallel Testing do not appear in the canvass is obviously appropriate.

If polling places are so small that there is no room to select one machine from the machines that were delivered to that polling place, it is possible to conduct Parallel Testing elsewhere, pulling machines for testing immediately prior to delivery to the polling place and setting them aside for testing. In that case, it is appropriate to publish the location of the testing and invite public observation. Casual drop-in observation can be maximized by conducting the tests near a polling place and advertising to the voters at that polling place that they can stop by after voting to watch or perhaps participate.

Post-election Testing

Some jurisdictions require routine post-election testing of some of the voting machinery, to make sure that, after the canvassing process was completed, the machinery is still working as well as it did before the election. Generally, these tests are very similar to pre-election or Logic and Accuracy testing.

Clearly, where the machines themselves hold the evidence of the vote count, as with mechanical lever voting machines or direct recording electronic voting machines, this evidence must not be destroyed until law and prudence agree that it is no longer relevant to any potential legal challenge to the election.

In the event of a recount, all of the pre-election tests that do not involve possible destruction of the votes being recounted must be repeated in order to assure that the machinery used in the recount is operating correctly.

APPENDIX F

EXAMPLE OF TRANSPARENT RANDOM SELECTION PROCESSES

A transparent random selection is one where members of the public can verify that, at the time of the choice, all selections were equally probable. Here are two examples of (reasonably) transparent random choice methods. There are many variations on these methods.

Method A: Each member of a group of individuals representing diverse interests chooses a random number (by any method) in a specified range 1...N and writes it down on a slip of paper. After each participant has chosen a number, the numbers are revealed to all and added. They are then divided by N, and the "integer remainder" is the number that is chosen (this is known in mathematics as the "modulo").

The best way to understand this is by example. Little Pennasota County has 9 machines (labeled "1" through "9") and wants to select one of these machines to parallel test. They want to ensure that the machine is chosen at random. To do this, they bring together several participants: a member of the League of Women Voters, the Democratic-Republicans, the Whigs, the Green Party, and the Libertarian Party. Each person is asked to select a number. The League of Women Voters' representative selects the number 5, the Democratic-Republican chooses 6, the Whig chooses 9, the Green chooses 8 and the Libertarian chooses 9. These numbers are then revealed and added: 5+6+9+8+9=37. They are then divided by 9. The "integer remainder" is 1, because 37 is divisible by 9 four times, with an "integer remainder" of 1 (or, 36 + 1). In this scenario, machine number 1 is chosen.

Any member of the group can assure the result is not "fixed" by the others. In the example above, all of the political parties might want to conspire to ensure that machine number 2 is picked for Parallel Testing. However, the League of Women Voters representative will prevent them from being able to do this: without knowing what number she is going to pick, they cannot know what the integer remainder will be.

Method B: Color-coded, transparent 10-sided dice are rolled (in a dice cup) in public view. The digits on the top faces of the dice are read off in a fixed order determined by the colors (*e.g.*, first red, then white, then blue). This yields a random 3-digit number. If the number is out of the desired range, it is discarded and the method performed again.

Note about transparently random selection process:

For a transparently random selection process to work, (1) how the randomly selected number is going to be used must be clearly stated in advance (*i.e.*, if we are choosing a number to decide which machine to parallel test, each machine must be labeled with one of the numbers that may be chosen), (2) the process for randomly selecting numbers must be understood by all participants, and (3) the event of randomly selecting numbers must be observable to all participants (and, if possible, members of the public).

For example, if we are picking what team of police are going to be left to look after the locked-up and security-sealed election materials before completion of the Automatic Routine Audit, the observers and participants must see the committed list of police that are being selected from in advance of the selection. The list must be posted visibly or in some other way "committed to" so that the association between random numbers selected and people selected cannot be switched after the numbers are produced.

In terms of assigning auditors to roles and machines to be audited, the goal might be to make sure that there is one Democratic-Republican and one Federalist assigned to review the paper records (the readers) and one Democratic-Republican and one Federalist assigned to tally the records (the writers). There should be no way to know what machines anyone will be assigned to, nor who will be teamed with whom during the audit.

If the use or interpretation of the random numbers is not clear and committed in advance, then an appropriately situated attacker might "interpret" the random number in a way that allows the attack go undetected by, for example, assigning attackers as auditors for all the subverted machines.[206]

APPENDIX G

ASSUMPTIONS

FACTS/ ASSUMPTIONS ABOUT THE PENNASOTA GOVERNOR'S RACE REFERRED TO IN THIS REPORT

GENERAL FACTS/ASSUMPTIONS ABOUT PENNASOTA IN 2007

Total Number of votes cast in gubernatorial election	3,459,379
Votes Cast for Tom Jefferson	1,769,818
Votes Cast for Johnny Adams	1,689,561
Margin of victory (votes) for Tom Jefferson	80,257
Margin of victory (%) for Tom Jefferson	2.32%

Target % votes to change in favor of Adams	3.0%
Target votes to add or subtract in hypothetical attacked election	103,781
Target votes to switch in Governor's Race	51,891

Total Number of Polling Places in State	3,030
Total Number of Precincts/Election Districts in State	4,820

LIMITS ON ATTACKER

Maximum % Votes Added or Subtracted Per County:	10% (5% switch)
Maximum % Votes Added or Subtracted Per Polling Place:	15%(7.5% switch)
Maximum % of Votes Added or Subtracted Per Voting Machine	30% (15% switch)

FACTS/ASSUMPTIONS ACROSS SYSTEMS

Minimum Number counties attacked	3
Total Number of polling places in State	3,030
Number of votes / polling place	1,142
Number polling stations that must be attacked where less than 15% of votes are added or subtracted	606

Minimum Number to develop and install Trojan Horse	1
Minimum Number to parameterize Trojan Horse	1

Number of machines out per polling place to create "bottleneck"	3
Maximum number of discouraged voters (decide not to vote) per polling place under bottleneck	88 (7.7%)
Number of votes potentially gained at polling station under bottleneck	70

Maximum % unfriendly voters in targeted polling places under bottleneck	90%
Percentage of friendly – foe votes under bottleneck	10%

Number Observers of Polling Book	1
Number people need to delete voter from poll book per polling place	1
Number Required To Modify enough Poll Books to change outcome of statewide election	606

Number of times single person can fraudulently vote	10

Number required to subvert audit	386

GENERAL ASSUMPTIONS FOR THREE LARGEST COUNTIES IN PENNASOTA: MEGA, CAPITAL AND SUBURBIA

Number of polling places in 3 largest counties	1,133
Number of precincts/Election Districts in 3 largest counties	1,669
Number of votes in 3 largest counties	1,156,035
Number of votes stored at largest tally center	531,584
Number of votes stored at the second largest tally center	360,541
Number of votes stored at third largest tally center	263,936
% votes that would need to be switched in the 3 largest counties to change outcome of governor's race	4.49%

VVPT-RELATED ASSUMPTIONS

Number of votes per DRE w/ VVPT	120
Number DREs w/ VVPT in State	28,828
Number DREs w/ VVPT in 3 Largest Counties	9634
Number of VVPT that must be changed to win election (assuming no more than 30% of votes switched on any roll)	2,934

Number required to create fake VVPT printouts to be replaced after polls close	3

PCOS AND BMD-RELATED ASSUMPTIONS

Total Number of PCOS machines in state	4,820
Votes per PCOS machine	606
Total Number of PCOS Machines in 3 Largest Counties	1,669

Number required to replace ballots with counterfeits per polling place	1
Number required to replace sufficient ballots	

with counterfeit complete ballots	606
Number required to steal or counterfeit ballot paper	5

DRE-RELATED ASSUMPTIONS

Number DREs in State	27,675
Number DREs in 3 largest counties	9,248
Number of votes per DRE	125
Number machines under Parallel Testing	58
Number required to subvert Parallel Testing	58
Maximum Number of votes switched on DRE	18.75
Minimum Number DREs attacked to swing election	2817

AUDIT ASSUMPTIONS

Number of Votes Audit Team Can Audit in One Day	120
Number of Auditors Per Team	2
Number of Votes Audited in 3 Largest Counties (2% audit)	23,121
Number of Audit Teams to Conduct Audit in 3 Largest Counties in one day	193
Total Auditors in 3 Largest Counties	386

APPENDIX H

TABLES SUPPORTING PENNASOTA ASSUMPTIONS

PENNASOTA COMPOSITE FROM VOTES IN THE 2004 BATTLEGROUND STATES (TAKEN FROM ACTUAL 2004 PRESIDENTIAL VOTE)

State	Total Votes for Adams (Kerry)	Total Votes for Jefferson (Bush)	Largest Three Counties in State by Population (in descending order)	Number of Votes for Adams (Kerry) by County	Number of Votes for Jefferson (Bush) by County
Colorado	1,001,725	1,101,256	Denver	166,135	69,903
			El Paso	77,648	161,361
			Jefferson	126,558	140,644
Florida	3,583,544	3,964,522	Miami-Dade	409,732	361,095
			Broward	453,873	244,674
			Palm Beach	328,687	212,688
Iowa	741,898	751,957	Polk	105,218	95,828
			Linn	60,442	49,442
			Scott	42,122	39,958
Michigan	2,279,183	2,313,746	Wayne	600,047	257,750
			Oakland	319,387	316,633
			Macomb	196,160	202,166
Minnesota	1,445,014	1,346,695	Hennepin	383,841	255,133
			Ramsey	171,846	97,096
			Dakota	104,635	108,959
Nevada	397,190	418,690	Clark	281,767	255,337
			Washoe	74,841	81,545
			Carson	9,441	13,171
New Mexico	370,942	376,930	Bernalillo	132,252	121,454
			Dona Ana	31,762	29,548
			Santa Fe	47,074	18,466
Ohio	2,741,165	2,859,764	Cuyahoga	448,503	221,600
			Franklin	285,801	237,253
			Hamilton	199,679	222,616
			Philadelphia	542,205	130,099

Pennsylvania	2,938,095	2,793,847	Allegheny	368,912	271,925
			Montgomery	222,048	175,741
			Milwaukee	297,653	180,287
Wisconsin	1,489,504	1,478,120	Dane	181,052	90,369
			Waukesha	73,626	154,926
Total Votes Per Candidate (2.32% margin of victory)	1,769,818	1,689,561	Average Votes of Three Largest Counties	674,295	481,767
Average Total Votes Per Candidate	3,439,379				

SOURCES: 2004 PRESIDENTIAL ELECTION VOTE TOTALS

Colorado
County: http://www.census.gov/popest/counties/tables/CO-EST2004-01-08.xls
elections: http://www.elections.colorado.gov/WWW/default/Prior%20Years%20Election%20Information/2004/Abstract%202003%202004%20082305%20Late%20PM-5.pdf
Florida
County: http://www.stateofflorida.com/Portal/DesktopDefault.aspx?tabid=95#27103
Election: http://election.dos.state.fl.us/elections/resultsarchive/Index.asp?ElectionDate=11/2/04&DATAMODE=
http://www.cnn.com/ELECTION/2004//pages/results/states/FL/P/00/county.000.html
Idaho
http://www.census.gov/popest/counties/tables/CO-EST2004-01-16.xls
http://www.idsos.state.id.us/ELECT/RESULTS/2004/general/tot_stwd.htm
http://www.idsos.state.id.us/ELECT/RESULTS/2004/general/cnty_pres.htm
Michigan
http://www.census.gov/popest/counties/tables/CO-EST2004-01-26.xls
http://miboecfr.nicusa.com/election/results/04GEN/01000000.html
Minnesota
http://www.census.gov/popest/counties/tables/CO-EST2004-01-27.xls
http://electionresults.sos.state.mn.us/20041102/
Wisconsin
http://www.census.gov/popest/counties/tables/CO-EST2004-01-55.xls
http://165.189.88.185/docview.asp?docid=1416&locid=47
Pennsylvania
http://www.census.gov/popest/counties/tables/CO-EST2004-01-42.xls
http://www.electionreturns.state.pa.us/ElectionReturns.aspx?Control=StatewideReturnsByCounty&ElecID=1&OfficeID=1#P
Ohio
http://www.census.gov/popest/counties/tables/CO-EST2004-01-39.xls
http://www.sos.state.oh.us/sos/ElectionsVoter/results2004.aspx?Section=135
Nevada
http://www.census.gov/popest/counties/tables/CO-EST2004-01-32.xls
http://www.cnn.com/ELECTION/2004/pages/results/states/NV/P/00/county.000.html
New Mexico
http://www.census.gov/popest/counties/tables/CO-EST2004-01-35.xls
http://www.cnn.com/ELECTION/2004/pages/results/states/NM/P/00/county.000.html

AVERAGE VOTES FOR THE THREE LARGEST COUNTIES IN THE 2004 BATTLEGROUND STATES

Composite Counties	Adams (Kerry)	Jefferson (Bush)
Mega County	336,735	194,849
Capital County	202,556	157,985
Suburban County	135,003	128,934
Total of Averages	674,295	481,767

PENNASOTA COMPOSITE OF POLLING PLACES AND PRECINCTS IN THE 2004 BATTLEGROUND STATES

State	County	Number of Polling Places (Nov 2004 elections unles otherwise indicated)	Number of Precincts November 2004	Number of Polling Places Statewide	Number of Precincts Statewide
Colorado	Denver	288	422	2,318	3,370
	El Paso	185	378		
	Jefferson	323	330		
Florida	Miami-Dade	534	749	5,433	6,892
	Broward	520	777		
	Palm Beach	420	692		
Iowa	Polk	180	183	1,916	1,966
	Linn	85	86		
	Scott	63	63		
Michigan	Wayne	670	1,198	3,890	5,235
	Oakland	432	549		
	Macomb	259	383		
Minnesota	Hennepin	431*	430	3,750**	4,108
	Ramsey	178	178		
	Dakota	137	137		
New Mexico	Bernalillo	162****	413****	612	684
	Dona Ana	78	108		
	Santa Fe	50	86		
Nevada	Clark	329	1,042	526	1,585
	Washoe	118	250		
	Carson	2	26		

Ohio	Cuyahoga	584	1,436	6,602	11,366
	Franklin	514	788		
	Hamilton	593	1,013		
Pennsylvania	Philadelphia	1,637	1,681	4,000	9,432
	Allegheny	1,214	1,214		
	Montgomery	407	407		
Wisconsin	Milwaukee	N/A ***	N/A***	1,253	3,563
	Dane				
	Waukesha				
Statewide Average of 10 States				2,969	4,820

SOURCE

Unless otherwise indicated, information is from the data tables at the EAC *2004 Election Day Survey* available at http://www.eac.gov/election_survey_2004/state_data.htm.

* 341 as of June 29, 2005. Telephone interview with Hennepin County Elections Board representative (November 7, 2005).

** Figure is estimated. Telephone interview with Minnesota Secretary of State representative (February 21, 2005).

***Number of Precincts and Polling Places N/A because elections are administered at municipality level and data were not centralized at county level. Milwaukee City, the largest municipality in Milwaukee County has 202 polling places. Telephone interview with Milwaukee County Election Commission representative (November 7, 2005).

****Telephone interview with Bernalillo County Clerk's Office representative (November 14, 2005).

AVERAGE NUMBER OF PRECINCTS AND POLLING PLACES FOR THE THREE LARGEST COUNTIES IN THE 2004 BATTLEGROUND STATES

Composite Counties	Precincts	Polling Places
Mega County	502	839
Capital County	347	481
Suburban County	250	349
Total of Averages	1,099	1,669

APPENDIX I

December 7, 2005
From: Professor Henry Brady, University of California, Berkeley
To: The Task Force

Denial of the Vote: You asked what the typical distribution of spreads was in precincts. I've gone to two data sets that were readily at hand – Broward and Palm Beach County Florida for the 2000 Presidential race. These are both heavily democratic counties. Roughly Broward was 67% for Gore and Palm Beach was 60% for Gore.

Here are the frequencies by precinct "binned" into 10 intervals from 0% to 100% voting for Gore:

GOREPCC1—BROWARD COUNTY FLORIDA, 2000 PRESIDENTIAL — % GORE VOTE

	Bin Number	% Voting for Gore	Frequency	% of Precincts	Valid %	Cumulative %
Valid	1.00	0-10%	13	1.7	1.7	1.7
	2.00	10-20%	2	.3	.3	2.0
	3.00	20-30%	3	.4	.4	2.4
	4.00	30-40%	15	1.9	2.0	4.4
	5.00	40-50%	73	9.3	9.8	14.2
	6.00	50-60%	132	16.8	17.7	31.9
	7.00	60-70%	217	27.6	29.0	60.9
	8.00	70-80%	124	15.8	16.6	77.5
	9.00	80-90%	87	11.1	11.6	89.2
	10.00	90-100%	81	10.3	10.8	100.0
Total			747	95.2	100.0	
MissingSystem			38	4.8		
Total			785	100.0		

GOREPCCT—PALM BEACH COUNTY FLORIDA — 2000 PRESIDENTIAL—% GORE VOTE

	Bin Number	% Voting for Gore	Frequency	% of Precincts	Valid %	Cumulative %
Valid	1.00	0-10%	7	1.1	1.1	1.1
	2.00	10-20%	8	1.3	1.3	2.4
	3.00	20-30%	5	.8	.8	3.3
	4.00	30-40%	42	6.7	6.8	10.1
	5.00	40-50%	123	19.6	20.0	30.1
	6.00	50-60%	150	23.9	24.4	54.5

	7.00	60-70%	123	19.6	20.0	74.5
	8.00	70-80%	64	10.2	10.4	84.9
	9.00	80-90%	52	8.3	8.5	93.3
	10.00	90-100%	41	6.5	6.7	100.0
Total			615	98.1	100.0	
Missing System			12	1.9		
Total			627	100.0		

Note that there are lots of precincts with 90% or higher Gore vote (10% in Broward and 6.5% in Palm Beach). These precincts are rather large (730 ballots cast on average in Broward and 695 ballots cast in Palm Beach).

Here are the Bush results for Palm Beach.

BUSHPCCT—PALM BEACH COUNTY FLORIDA 2000 PRESIDENTIAL % BUSH VOTE

	Bin Number	% Voting for Gore	Frequency	% of Precincts	Valid %	Cumulative %
Valid	1.00	0-10%	55	8.8	8.9	8.9
	2.00	10-20%	49	7.8	8.0	16.9
	3.00	20-30%	76	12.1	12.4	29.3
	4.00	30-40%	148	23.6	24.1	53.3
	5.00	40-50%	157	25.0	25.5	78.9
	6.00	50-60%	87	13.9	14.1	93.0
	7.00	60-70%	27	4.3	4.4	97.4
	8.00	70-80%	3	.5	.5	97.9
	9.00	80-90%	6	1.0	1.0	98.9
	10.00	90-100%	7	1.1	1.1	100.0
Total			615	98.1	100.0	
Missing System			12	1.9		
Total			627	100.0		

Note that there are a lot fewer precincts with high Bush vote – only about 2.1% with 80% or greater Bush vote. But, of course, Palm Beach was a very highly Democratic County. Here are the results for Broward:

BUSHPCC1—BROWARD COUNTY FLORIDA — 2000 PRESIDENTIAL — BUSH VOTE

	Bin Number	% Voting for Gore	Frequency	% of Precincts	Valid %	Cumulative %
Valid	1.00	0-10%	94	12.0	12.6	12.6
	2.00	10-20%	96	12.2	12.9	25.4
	3.00	20-30%	144	18.3	19.3	44.7
	4.00	30-40%	211	26.9	28.2	73.0

5.00	40-50%	122	15.5	16.3	89.3
6.00	50-60%	53	6.8	7.1	96.4
7.00	60-70%	11	1.4	1.5	97.9
8.00	70-80%	1	.1	.1	98.0
9.00	80-90%	2	.3	.3	98.3
10.00	90-100%	13	1.7	1.7	100.0
Total		747	95.2	100.0	
MissingSystem		38	4.8		
Total		785	100.0		

Note that we have about the same situation for Broward.

This suggests that it would be harder to do a "denial of the vote" for Bush than for Gore in these counties. But, of course, in a Presidential race you would probably first choose a county that was heavily in the direction of the other party – hence, if you were a Republican you would choose Palm Beach or Broward Counties and you would not choose heavily Republican counties in the North of Florida.

These tables are typical of what we see around the country.

APPENDIX J

CHANCES OF CATCHING ATTACK PROGRAM THROUGH PARALLEL TESTING

The Automatic Routine Audit and Parallel Testing should both use random sampling of precincts or voting machines to try to catch misbehavior. The attacker doesn't know ahead of time which precincts or machines will be checked and, if there are enough random samples taken, she cannot tamper with a substantial number of precincts or machines without a big risk of her tampering being caught. The question we address in this Appendix is how many machines must be randomly tested to reliably detect a certain level of tampering.

One way to visualize the way random sampling can work is to imagine a room full of ping pong balls. Most of the balls are blue, but a small fraction (say, 1/2 of 1%) are red. When we sample them, we reach into the bin without looking and draw out a ball; we want to know whether we are likely to draw out a red ball in a certain number of tries.

We can imagine a literal version of this, with each ball or slip of paper having a different machine or polling place ID on it. In the case of Parallel Testing, we select machines by drawing these balls out of the bin, and sampling only what is indicated by those balls. If we draw a ball representing a machine whose results have been tampered with, we will detect the tampering; if none of the tampered machines is tested, the attacker will get away with her tampering. This idea is very general–it can be applied to Automatic Routine Audits of polling places, precincts or voting machines, Parallel Testing of machines, careful physical inspection of tamper-evident seals on ballot boxes, inspection of polling places for compliance with election laws, *etc.*

The way we really do this is called "sampling without replacement," which just means that when we draw a ball out of the bin, we don't put it back. The probablities of finding the red ball changes each time we draw a ball out.. If we have a reasonably large number of balls in the bin, and we are sampling a small percentage, we can use a much simpler formula for sampling with replacement that's approximately correct. This binomial estimate will generally err in a conservative direction, *i.e.,* we will draw a sample larger than necessary.

It's easy to convince yourself that drawing more balls from this bin makes you more likely to get one of the rare balls. It is also easy to see that the more red balls there are in the bin, the more likely you are to draw one out.

We can write formulas to describe all this more precisely. Suppose that in Pennasota there are 28,828 DREs, and 2,883 (or 10%) have been tampered with.[207] We're going to test 10 machines. We want to know how likely we are to detect the tampering.

The easiest way to think of this is to ask how likely we are to fail to detect the tampering. (If we have a 10% chance of failing to detect the tampering, that's just another way of saying we have a 90% chance of detecting it.) Each time we draw a ball from the bin, we have approximately a (2,883/28,828) = 0.10 chance of getting a ball that represents one of the tampered machines. The probability that we'll fail to sample a tampered machine each time is approximately 0.90. To figure out what the probability is that we will fail to sample one of the tampered ones 10 times in a row, we just multiply the probabilities together: 0.90 * 0.90 * ... * 0.90 = $(0.90)^{10}$ = 0.35. So, after 10 samples, we have about a 35% chance of not having caught the attacker. Another way of saying the same thing is that we have about a 100% – 35% = 65% chance of catching the attacker.

An approximate formula for this is:

C = fraction compromised
N = number sampled
Probability[detect attack] = $1 - (1\text{-}C)^{s}$

Writing the probabilities as percentages, this looks like:

Probability[detect attack] = $100\% - (100\%\text{-}C)^{s}$

Now, the question we really care about is how many samples we must take to have some high probability of detecting an attack. That is, we may start knowing the P[detect attack] value we want, and need to work backward to find how many samples we must take if the attacker has tampered with 10% of our machines. The general (approximate) formula is

D = probability of detection
C = fraction compromised
N = number sampled

N = log(1-D)/log(1-C)

where log() is just the logarithm of these probabilities. The base of the logarithm doesn't matter.

Some sample values for this, with D = 95%. (That is, we require a 95% chance of catching the tampering.)

% Compromised	Number Sampled
1/2%	598
1%	298
2%	148
5%	58
10%	28
25%	10

This formula and table are approximate. For small numbers of machines or precincts being sampled, they overstate the number of samples needed to get the desired probability, which means that following them may lead you to be a little more secure than you need to be.

So even if we assume that only 5% of machines are tampered with, Parallel Testing of 58 machines should give us a 95% chance of catching a machine that has been tampered with.[208]

APPENDIX K

CHANCES OF CATCHING ATTACK PROGRAM THROUGH THE ARA

From the math already done in Appendix J, we can create this formula:

As already discussed, the formulas listed in Appendix J will apply just as well when attempting to determine whether a 2% audit will have a good chance of catching a fraud.

There are more than 28,000 DREs w/ VVPT in Pennasota, with an average of 120 voters per machine. As our attacker wants to avoid detection, we have assumed that she will create an attack program that will switch a limited number of votes in each polling place – specifically about 18 (or 15% of all votes) per machine. Assuming she wants to switch about 52,000 votes, this comes out to an attack on about 1600 machines.

What is the probability of catching this fraud with a 2% audit? In a 2% audit, we will audit about 560 machines.

The fraction of bad machines is 1,600/28,000 or 0.055.

Each time we audit a machine, we have a chance of 0.055 of picking a machine that has been tampered with, and a chance of 1-0.055 (or 0.945) of picking a machine that has not been tampered with.

The probability of picking *only* machines that have not been tampered with after auditing all 560 machines is $(1\text{-}C)^s$ or $(0.945)^{560}$. This is extremely close to zero, which means that the chances of *not* catching the fraud are less than 1%; conversely, the chances of catching it are close to 100%.

Paper replaced

But what if the attacker had pollworkers in 550 polling places replace the paper before it reached county headquarters for the ARA? This would leave, at a minimum 56 rolls that are evidence of the fraud (assuming that in the 56 polling places where paper wasn't replaced, there was only one DRE per polling site). This means roughly 0.2% of paper rolls would show that the paper did not match the electronic records. What are the chances that a 2% audit (or audit of 560 machines) would catch this?

This time, each time we audit the paper rolls, the chances of catching a paper roll with evidence of the fraud is 56/28,000, or roughly 0.002. So the probability of picking *only* rolls that do not show evidence of fraud after auditing all 560 rolls and machines is $(.998)^{560}$, or about 1/3. Thus, there would still be a 2/3 chance that the fraud would be detected.

APPENDIX L

Subverting the Audit/ Parallel Testing

We've described auditing processes that can detect all kinds of misbehavior. However, this leaves open a question: How many auditors must our attacker corrupt to prevent the detection of misbehavior?

Preliminaries

We assume that auditing or Parallel Testing is done by teams. Each team is somehow put together from one or more auditors, and each team is assigned randomly to a subset of the things being audited.

How Many Corrupt Auditors Subvert an Audit Team?

How many corrupt auditors does it take to subvert an audit team? The answer depends on the procedures used for auditing. The two extreme cases are of the greatest interest:

- One Bad Apple: As discussed in page 55 of this report, during Parallel Testing, it is likely that a single corrupt auditor can enter a Cryptic Knock that will inform a tampered machine that it is being parallel tested. If the tester cannot enter a Cryptic Knock (because this feature was not part of the attack program) then all members of the Parallel Testing team will have to be subverted.

- The Whole Bunch: During hand-recounts of paper ballots, reasonable procedures can make it very difficult for an audit team with even one uncorrupted auditor to fail to detect any significant fraud (that is, more than two or three votes).

We will consider these two models below.

Impact of Corrupted Audit Teams

The best way to think about the impact of a corrupt audit team is to omit the audits done by that team from the total number of audits we assume are done. Thus, if we have ten teams, each doing 5 audits, and we assume two teams are corrupt, then instead of calculating the probability of detecting an attack based on 50 audits being done, we calculate it based on the probability of 40 audits being done.

Some Simple Approximations

Here is a simple, conservative approximation of the 95% upper bound on the number of compromised audit teams:

We assume sampling without replacement. We compute the probability that a team will get corrupted, and then use binomial distribution to determine the expected number of corruptions. This is also oversimplified and conservative, but less so than the super-simple model above.

Let:

R be the total number of auditors, of whom N are corrupt.
Each team consist of K auditors
Q = R/K = the total number of teams

For the one corrupt auditor model:
(That is, a single corrupt auditor subverts the whole team.)
The probability of a team being corrupted is P = 1-((R-N)/R)K.

For the all corrupt model:
(That is, all the auditors on the team must be corrupt to corrupt the team.)
The probability of a team being corrupted is P = (N/R)K.

For both models:
Prob(M corrupted audit teams) = Choose(Q,M) P^{S} $(1-P)^{(Q-S)}$
Expected number of corrupted audit teams = P*Q
S = standard deviation = Sqrt(P*(1-P)*Q)
95% upper bound on corrupted audit teams = P*Q + 1.64*S

The biggest thing to notice about these formulas is that when you need to corrupt all members of a team to corrupt the team, you need to corrupt practically all the auditors to have much of an impact. For example, consider an election with 100 auditors, 5 to a team. Here are some numbers when we have to have all auditors on a team corrupted to subvert that team's audits: (There are 20 teams total.)

Corrupt	Exp. Corrupt Teams	95% Upper Bound
10	0	0
20	0	0
30	0	0
40	0	1
50	1	2
60	2	4
70	3	6
80	7	10
90	12	15

The 95% upper bound here means that the true number of corrupt teams should not exceed the upper bound in 95% of the possible teams drawn.

*Note the implications for parameters of our audit teams – bigger teams are much better than smaller ones. If we had audit teams of one, corrupting half the auditors would corrupt half the audits, while here it corrupts only 10% of the audits. On the other hand, we could do five times as many audits with one auditor to a team.

On the other hand, the attacker has a much easier time attacking auditing processes where a single corrupted participant subverts the whole audit process. Similar numbers then look like:

10	8	11
20	13	16
30	17	19
40	18	20
50	19	20
60	20	20
70	20	20

In this case, small audit/Parallel Testing teams make more sense.

Bribing The Audit Teams in Pennasota to Subvert the Audit

If our attacker could successfully bribe auditors to "cheat" during the audit, so that they would ignore discrepancies between the paper and electronic records, how many would he have to bribe? Our analysis shows that nearly all of the auditors in the largest counties would have to be successfully bribed if the attack was to work.

We can use the audit in Pennasota's three largest county, Mega, Capitol and Suburbia, as an example. With a 2% audit, 193 teams of two will audit one DRE w/ VVPT paper roll each (each paper roll will contain approximately 120 votes). Each member of each team of auditors is selected by one of the major political parties; after they are selected and immediately before the auditing begins, they are randomly assigned a partner and a machine. Every team has one Federalist and one Democratic-Republican.

What fraction of these auditors must the attackers corrupt to avoid her attack being caught? If τ represents the fraction of auditors from each party that our attacker must corrupt, and each party's auditor is randomly matched with an auditor from the other party, the probability of an entire audit team being corrupted (i.e. both auditors being corrupted) is τ^2.

A machine passes an audit if

(1) it is a good machine; or
(2) it is a bad machine but both auditors are corrupted.

The probability of (1) is 1- C. The probability of (2) is $C\tau^2$ Thus the probability of a machine passing the audit is

$$1 + C\,(\tau^2 - 1).$$

And the probability of S machine passing the audit is approximately

$$\rho = (1 + C\,(\tau^2 - 1)^S.$$

Solving this equation for τ yields

$$\tau = \sqrt{\frac{\rho^{(1/S)} - 1}{C} + 1}$$

We have assumed that the attacker would need to attack 1,602 DREs w/ VVPT to feel comfortable that he could change the outcome of the Governor's race in Pennasota. There are 9,634 DREs w/ VVPT in Pennasota's three largest counties. Thus, C=1602/9634 or 0.17. S, the number of machines and paper rolls audited is 193. Assuming that our attacker wants 90% certainty that she will subvert the audit, ρ equal 0.9.

Accordingly, the percentage of auditors that must be successfully bribed to subvert the audit is close to approximately 99.7%.

APPENDIX M

The following are examples of procedures that would allow jurisdictions to respond effectively to detection of bugs or Software Attack Programs:

1. impound and conduct a transparent forensic examination[209] of all machines showing unexplained discrepancies during Parallel Testing;

2. where evidence of a software bug or attack program is subsequently found (or no credible explanation for the discrepancy is discovered), conduct a forensic examination of all DREs in the state used during the election;

3. identify the machines that show evidence of tampering or a software flaw that could have affected the electronic tally of votes;

4. review the reported margin of victory in each potentially affected race;

5. based upon the (a) margin of victory, (b) number of machines affected, and (c) nature and scope of the tampering or flaw, determine whether there is a substantial likelihood that the tampering or flaw changed the outcome of a particular race; and

6. where there is a substantial likelihood that tampering changed the outcome of a particular race, hold a new election for the office.

The following are examples of procedures that would allow jurisdictions to respond effectively to detection of statistical anomalies in the voter-verified paper record:

1. Conduct a transparent forensic investigation of machines[210] that have produced paper records with significant statistical anomalies;

2. To the extent tampering with any of these machines is found, conduct a similar investigation of all machines in the State;

3. After quantifying the number of machines that have been tampered with, determine the margin of victory in each potentially affected race;

4. Based upon the (a) margin of victory, (b) number of machines affected, and (c) nature and scope of the tampering, determine whether there is a substantial likelihood that tampering changed the outcome of a particular race; and

5. In the event that a determination is made that there is a substantial likelihood that tampering changed the outcome of a particular race, hold a new election for the office.

ENDNOTES

[1] Ballot Marking Devices have been purchased by several jurisdictions in recent months. However, they have not yet been purchased as the primary machine in any jurisdiction's voting system. Instead, they have generally been purchased as the "accessible" unit, to meet the Help America Vote Act's accessibility requirements. Lawrence Norden, *Voting System Usability in* The Machinery of Democracy (Brennan Center for Justice ed., forthcoming July 2006).

[2] These systems are currently used to a limited extent in both Vermont and New Hampshire. Lawrence Norden *et al., Voting System Accessibility, in* The Machinery of Democracy (Brennan Center for Justice ed., forthcoming June 2006).

[3] These systems are currently in development and not commercially available. They are discussed in further detail *infra* at p. 92.

[4] In 2004, 27 States allowed early voting. Approximately 19.3% of voters in these states voted early. Approximately 11.6% of votes counted in 2004 were absentee ballots. Oregon is the only state with an all-mail voting system. *See* Election Assistance Commission, *EAC Election Day Survey,* http://www.eac.gov/election_survey_2004/statedata/StateLevelSummary.htm (turnout source tab at bottom) (Last visited May 25, 2006).

[5] These reports will be released under separate cover in 2006. *See supra* notes 1 and 2 and *infra* note 184.

[6] NIST has informed the Brennan Center that the development of policy recommendations for voting systems is not within the agency's mission or institutional authority. Accordingly, the policy recommendations in the report should not be attributed to Task Force members who work for NIST.

[7] Tracy Campbell, Deliver the Vote, at xvi (2005) (pointing to, among other things, a history of vote buying, ballot stuffing, and transposing of results).

[8] *Id.*

[9] Joseph P. Harris, Election Administration in the United States (1934).

[10] *See e.g.* Deliver the Vote, *supra* note 7 at 275-284; Edmund F. Kallina, Jr., Courthoue over White House – Chicago and the Presidential Election of 1960 (1988) (documenting fraud found in Chicago's 1960 elections); Andrew Gumbel, Steal this Vote, at 173-200 (2005) (detailing tampering and questionable results in the era of lever and punch-card voting).

[11] Deliver the Vote, *supra* note 7 at 83, 99, 137.

[12] *See, e.g., Chip Glitch Hands Victory to Wrong Candidate,* Associated Press, Nov. 11, 2002 (noting that a "defective computer chip in [Scurry] County's optical scanner misread ballots . . . and incorrectly tallied a landslide victory for Republicans.")

[13] *See, e.g., Computer Loses More Than 4,000 Early Votes in Carteret,* Charlotte Observer, Nov. 4, 2004 (noting that as a result of a software bug, machines could only store 3,005 votes; after this number of votes was recorded the machines accepted, but did not store, the ballots of 4,438 voters in the 2004 presidential election).

[14] *See, e.g.,* Anna M. Tinsley and Anthony Spangler, *Vote Spike Blamed on Program Snafu,* Fort Worth Star-Telegram, Mar. 9, 2006, (noting that a programming error in the tally server software caused an extra 100,000 votes to be initially recorded in Tarrant County, Texas)

[15] *See, e.g.,* Susan Kuczka, *Returns Are In: Software Goofed – Lake County Tally Misled 15 Hopefuls,* Chicago Tribune, Apr. 4, 2003, at 1 (noting that programming error caused machines to record names of wrong candidates)

[16] *See, e.g., Voters Turned Away After Waiting Hours* (WPLG Local 10 News television broadcast, Nov. 1, 2004) (noting that breakdowns of DREs in Broward County forced people to wait to vote

for hours before they could vote), available at http://www.local10.com/news/3878344/detail.html.

[17] *See, e.g.,* Kevin P. Connolly, *Computer Glitches Slow Volusia Results: County Officials Ask the Machine's Supplier to Investigate Why Memory Cards Failed Tuesday,* ORLANDO SENTINEL, Nov. 4, 2004 at A17.

[18] *Nearly 40 Votes May Have Been Lost in Palm Beach County,* USA TODAY, Nov. 2, 2004, at B7 (noting that failure to properly plug in machine appeared to cause the loss of as many as 40 votes).

[19] Douglas W. Jones, *Threats to Voting Systems* at 2 (Oct. 7, 2005), *available at* http://vote.nist.gov/threats/ papers/threats_to_voting_systems.pdf (presented at the NIST Threat Analysis Workshop).

[20] The annexed catalogs are available at www.brennancenter.org [hereinafter *Attack Catalogs*].

21 We determined that looking at each attack in the context of an effort to change a statewide election was critical to determining its difficulty. There are many ways to switch or spoil a single vote. It would be impossible for election officials to guard against all such threats. The challenge is to prevent those attacks that (a) are feasible, and (b) if carried out successfully would affect a large number of votes. By looking at attacks that could affect statewide elections, we have attempted to limit ourselves to these types of attacks.

[22] *See, Attack Catalogs, supra* note 20.

[23] The specifics might differ slightly. A vote buying scheme against DREs or DREs with VVPT could involve the use of a small camera, whereby the voter would photograph the confirmation screen or VVPT to prove that she voted the way she promised. This would not work in the case of a PCOS vote, as there is no display confirming the voter's intention. To merely take a picture of the PCOS ballot would prove nothing – the voter could photograph a ballot that showed she voted for Johnny Adams, but erase that vote and submit his ballot marked for Tom Jefferson. *See* Attack Number 26 in the DRE w/ VVPT Catalog and Attack Number 26 in the DRE Catalog, *Attack Catalogs, supra* note 20.

[24] Of course, statewide elections are occasionally decided by mere dozens or hundreds of votes. But these are the exceptions among the exceptionally close races. As discussed in more detail, *infra* at pp. 20–23, we have assumed that in attempting to affect a close statewide race, an attacker must presume that one candidate's margin of victory will be somewhere from 2–3% of all votes.

[25] *See* PCOS Attack Catalog, Attack Catalogs, *supra* note 20.

[26] In assigning values, we have made certain assumptions about the jurisdiction's security measures. As discussed in greater detail, *infra* at pp. 14–15, these assumptions are based upon survey responses from and interviews with current and former election officials about their security practices. Among the assumptions we have made: (1) at the end of an Election Day, but prior to the transportation of ballots, poll workers check the total number of votes cast against the poll books in each polling place, and (2) ballots from each polling place are delivered to central county offices separately (*i.e.,* a single person or vehicle does not go from polling place to polling place collecting ballots before delivering them to the central location).

[27] This number was reached after considering the total number and types of ballots that would have to be stolen or created.

[28] Given the difficulty of stuffing the ballot box and modifying poll books, we have assumed that at least one person would be needed for each task in every polling place where it is accomplished. Of course, there is a real possibility that if this attack were carried out, someone would get caught. At the very least, stuffing the ballot box and modifying the ballot boxes *in the polling place* would be difficult to do without attracting notice. If anything, this fact supports our methodology. It is not impossible to imagine that, with the proper motivation and skills, two people could accomplish these goals in a single polling place somewhere in the country. It is far more difficult to imagine dozens or hundreds of people accomplishing this task successfully in dozens or hundreds of

polling places in the same state. For this reason, and under our methodology, the attack is labeled "very difficult" to accomplish successfully.

[29] Among those interviewed in July and Aug. of 2005 regarding the difficulty of various attacks on election systems were Debbie Smith, Elections Coordinator, Caleveras County, CA; Patrick F. Gill, Auditor, Sioux City, IA; Wendy Noren, County Clerk of Boone County, MO; Beverly J. Harry, County Clerk/Registrar of Voters, Inyo County, CA ; Larry Lomax, Registrar of Voters, Clark County, NV; Cliff Borofsky, Election Administrator for Bexar County, TX; F. Robert Williams, Chief Information Officer for Monmouth County, NJ; and Brian Newby, Election Commissioner of Johnson County, KS.

[30] Wikipedia, *US Senate Election, 2000*, http://en.wikipedia.org/wiki/U.S._Senate_election,_2000 (as of May 25, 2006, 15:30 GMT)

[31] International Information Programs, *2004 U.S. Elections Results Finally Complete*, http://usinfo.state.gov/dhr/Archive/2005/Jan/03-462014.html (Dec. 30, 2004).

[32] Zogby International, *Election 2004 Zogby Battleground State Polls, at* http://www.zogby.com/news/ ReadNews.dbm?ID=904 (Oct. 24, 2004).

[33] While our results are derived from a review of a composite election in a composite jurisdiction, we believe they are applicable to similarly close elections in almost any state. As a check on our findings, we have run an analysis of Attack Catalogs against the Presidential race in Washington State in 2004, and come up with substantially similar results to those discussed in this paper.

[34] Steganography is "the art and science of writing hidden messages in such a way that no one apart from the intended recipient knows of the existence of the message." Wikipedia, *Steganography,* http://en.wikipedia.org/wiki/Steganography (as of May 25, 2006, 15:33 GMT).

[35] *See infra* note 122.

[36] Responses to the Brennan Center Security Survey are on file at the Brennan Center. For a sample survey, *see* Appendix D.

[37] Starting with the 1990 FEC/NASED standards, Independent Testing Authorities ("ITAs") have tested voting systems, certifying that these systems meet the letter of the "voluntary" standards set by the federal government and required, by law, in most states. Several states, such as Florida, that impose additional standards contract with the same labs to test to these stronger standards. In the future, the EAC will be in charge of certification that will be done by VSTLs (Voting System Test Labs). For further explanation of this change, *see* Election Assistance Commision, *Voluntary Voting System Guidelines* (2005), *available at* http://www.eac.gov/VVSG%20Volume_II.pdf (Last visited May 31, 2006). For further discussion of the testing most machines undergo, *see* Appendix E.

[38] Our analysis shows that this is a very important countermeasure. Specifically, this countermeasure allows pollworkers and the public to ensure that corrupt or flawed software on a county's central tally- server does not incorrectly add up machine vote totals.

[39] A thorough discussion of the types of testing voting machines might be subject to is provided in Appendix E

[40] We have assumed that each machine delivered by a vendor to the jurisdiction is tested by that jurisdiction. Even if the vendor has some kind of quality control guarantees, these are of no value unless the customer detects failures at the time of delivery. At minimum, such tests would include power-on testing, basic user interface tests (do all the buttons work, does the touch-screen sense touches at all extremes of its surface, do the paper-feed mechanisms work, does the uninterruptible power supply work). This is known as "Acceptance Testing." For a more detailed discussion of Acceptance Testing, refer to Appendix E.

[41] We have assumed that before each election every voting machine would be subject to public testing. This is frequently described as Logic and Accuracy testing or simply L&A testing, a term

that is more appropriate in the realm of punch-card and mark-sense ballot tabulating machines than in the realm of DRE systems, but the term is used widely and in many states it is enshrined in state law. For a more detailed discussion of Logic and Accuracy testing refer to Appendix E.

[42] Electionline.org, *Recounts: From Punch Cards to Paper Trails,* at 3 (Oct. 2005) [hereinafter *Recounts*], at http://www.electionline.org/Portals/1/Publications/ERIPBrief12.SB370updated.pdf (Last visited May 25, 2006).

[43] California selects auditors at the county level by political party. Telephone Interview by Eric L. Lazarus with Debbie Smith, Elections Coordinator, Caleveras County, CA (July 14, 2005). We assume each audit team will have at least two members, with one member selected by each political party.

[44] This might be difficult in the selection of machines for Parallel Testing. If election officials insist on one-month's notice as to which precincts will be tested, publication of the selected machines could be problematic. Specifically, this would allow an attacker to know which Precincts to avoid attacking.

[45] Many more recommendations for a sound Parallel Testing regime can be found in the subsection entitled "Effects of Regimen for Parallel Testing," *infra* at pp.52–61.

[46] In California election officials generally felt they needed at least a month's notice – this is because when Parallel Testing is done, certain precincts will lose the use of one or two machines. Telephone interview by Eric L. Lazarus with Jocelyn Whitney, Developer and Project Manager for Parallel Testing in California (Dec. 23, 2005).

[47] In a threat paper entitled *"Trojan Horse in DRE -OS"* posted by Chris Lowe for the NIST Threat Analysis Workshop in Oct. 2005, Mr. Lowe imagined an attack in an election involving Tom Jefferson and John Adams. The analysis in this paper should not be confused with Mr. Lowe's work, although we do reference Mr. Lowe's threat paper, *infra* note 120.

[48] Because this report does not address security issues related to absentee voting, and for purposes of simplicity, we are assuming that all votes were cast at a polling place on one of the three voting systems we are examining.

[49] The numbers in this appendix represent the average number of polling places and precincts in the three largest counties in each of the Zogby battleground states in 2004 presidential election (*See, supra* note 34). Milwaukee County was not included in this analysis because they divide up polling places and precincts in a way that made comparison impossible.

[50] If an attacker were to switch 4% of the votes from Candidate A to Candidate B, it would have the same effect on the margin of victory as adding 8% of the total votes to Candidate A, or subtracting 8% of the total votes from candidate B. This can be demonstrated in a simple example. Suppose Candidate A and Candidate B each received 50 votes. If we switched 4 votes from Candidate B to Candidate A, Candidate A would win the election by 8 votes: 54 for Candidate A, 46 for Candidate B. If on the other hand, we simply stuffed the ballot box and added 8 votes for Candidate A, but did not otherwise tamper with the election results, Candidate A would again win by 8 votes: 58 votes for Candidate A, and 50 votes for Candidate B.

[51] This assumes that the county does not post PDF images of the ballot on the web prior to the election; this was done by, among other counties, St. Lucie County, Florida prior to the General Election of 2000.

[52] *Id; see also* Appendix G

[53] This analysis does not even consider how much *more* difficult the attack would become if one of our two other sets of countermeasures was in place. For instance, under the Basic Set of Countermeasures, "ballot boxes are examined (to ensure they are empty) and locked by poll workers immediately before the polls are opened." This simple countermeasure would make PCOS Attack 12 significantly more difficult to execute successfully; the attackers could not simply scan ballots just before Election Day and hope that these ballots would become part of the tally. They would

have to co-opt every person charged with reviewing the ballot boxes prior to opening in all 606 targeted polling places.

[54] Cook County Election Department, *Results from November 2004 Elections, at* www.voterinfo net.com/ results/detail/summary.php?election=20041102G (Last visited May 31, 2006)

[55] Of course, it is possible that an attacker could switch more than this percentage of votes in a single machine, polling place or county without detection. To the extent that she could do so, her ability to successfully change the outcome of a statewide election would be made easier. For a complete list of assumptions made about Pennasota, *see* Appendix G.

[56] As discussed in greater detail, *infra at* p. 72, for some attack scenarios, the ability to carry out the attack in the fewest possible counties is key to (a) involving the fewest number of informed participants and (b) increasing the chances that the attack will not be detected. In other scenarios, a statewide attack is more likely to accomplish these goals.

[57] Specifically, our attacker would need to add or subtract less than six percent (6%) of votes in these three counties; this means she would need to "switch" (*i.e.,* move a vote from one candidate to another) less than three percent (3%) of votes in these counties.

[58] Based upon composite results from the three largest counties in each of the ten Zogby Battleground States reviewed, *See, supra* Zogby note 32.

[59] The fact that we list these categories of attacks does not mean that we necessarily believe an attacker could successfully use these attacks to affect the outcome of our statewide election. As discussed, *infra at* p. 69, we have concluded that some attacks would certainly fail if attempted. In such cases, the *Catalogs* label such attacks "N/A" under the column "Number of Informed Participants."

[60] By "very difficult" we mean that it would require hundreds or thousands of informed participants; or, regardless of how many participants are involved, it would not affect enough votes to change the outcome of a close statewide race.

[61] Dr. Michael Shamos, *Paper Trail Boycott* (Oct. 5, 2005) (a NIST Threat Analysis workshop presentation summarizing the logistics of this attack). A more detailed description of the attack can be found at http://vote.nist.gov/threats/papers/papertraiboycot.pdf.

[62] This number is a high estimate. *See* Professor Benjamin Highton, *In Long Lines, Voting Machine Availability and Turnout,* 39 POLITICAL SCIENCE AND POLITICS 65, 67 (2006) (estimating that long lines in Franklin County, Ohio resulted in a 7.7% reduction in turnout in certain very large precincts).

[63] There are 2,969 polling places in Pennasota. *See* Appendix G.

[64] This section of the report borrows and relies heavily on *"Strategies for Software Attacks on Voting Machines,"* a white paper presented by John Kelsey of NIST at the NIST Threat Analysis workshop in Oct. 2005. This section does not cover the technical details and challenges of creating a successful software attack program in the same detail as Mr. Kelsey's paper. That paper can be found *at* http://vote.nist.gov/threats/papers/stategies_for_software_attacks.pdf.

[65] *See* Computer Crime Research Center, *Report America Under Attack, at* www.crime-research.org/news/2003/04/ Mess0301.html (Last visited May 31, 2006) (noting a record number of computer hackers attacking military and government systems); *see also* Scott A. Boorman and Paul R. Levitt, *Deadly Bugs,* CHICAGO TRIBUNE (MAGAZINE) May 3, 1987 *at* C19 (detailing, among other attacks, the planting of a software bug in the computer system of the Los Angeles Department of Water and Power in 1985, which made some of the utilities' important internal files inaccessible for a week); Edward Iwata, *Companies Stress Network Security,* USA TODAY, Oct. 2, 2001 at 3B (citing "security audits" by security firm Sanctum in which they successfully broke "into the networks of 300 organizations, including federal agencies, financial firms and airlines").

[66] *See* John Deutch *Off Line: At War with the Info-Terrorists,* THE OBSERVER, July 7, 1996 at 7 (the former Director of the Central Intelligence Agency cites attacks on computers and software to

divert funds from banks, embezzle funds and commit fraud against credit card companies); L.A. Lorek, *Internet Worm Disrupts Business,* SAN ANTONIO EXPRESS-NEWS (Texas), Jan. 28, 2003 at 1E (discussing "Slammer," a computer worm which attacked a hole in Microsoft software and prevented banks and airlines from performing basic operations).

[67] There is an extensive history of successful attacks against content protection systems, such as those created to protect digital media. *See generally* Wikipedia, Digital Rights Management, http://en.wikipedia.org/wiki/Digital_rights_management (detailing many such attacks) (as of May 26, 2006 15:39 GMT). For instance, in Oct. 1999 a teenaged Scandinavian high school dropout, Jon Lech Johansen, broke a much heralded DVD encryption scheme. *See* Wikipedia, *Content-Scrambling System,* http://en.wikipedia.org/wiki/Content_Scrambling_System (as of May 26, 2006 15:39 GMT).

[68] Special purpose cryptographic devices are created to protect key material, even when an attacker has control over the device doing the encryption. There have been a number of successful attacks against such devices. *See* Ross Anderson, Mike Bond, Jolyon Clulow & Sergei Skorobogotav, *Cryptographic Processors – A Survey,* UNIVERSITY OF CAMBRIDGE COMPUTER LABORATORY TECHNICAL REPORT NO. 641 (Aug. 2005), at http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-641.pdf, for an excellent history of some of these high-level attacks.

[69] *See e.g.,* Jaikumar Vijayan, *Security Product Flaws are Magnets for Attackers,* COMPUTER WEEKLY, *at* http://www.computerweekly.com/Articles/Article.aspx?liArticleID=201449&PrinterFriendly=true (Mar. 29, 2004) (noting the growing number of attacks against "the very products users invest in to safeguard their systems").

[70] For an example of this type of attack, *see* the discussion of Ron Harris's attack on video poker machines, *infra* note 148.

[71] Domain Name System (DNS) is a distributed database that stores mappings of Internet Protocol addresses and host names to facilitate user-friendly web browsing. *See* Ian Betteridge, *Security Company Warns About DNS Attacks,* eWeek.com at http://www.eweek.com/article2/0,,1782543,00.asp, (Apr. 5, 2005) (for discussion of DNS attacks).

[72] Dennis Callaghan, *Federal Sweep Nets Spammers, Cyber-Criminals,* eWeek.com, at http://www.eweek.com/print_article2/0,1217,a=134159,00.asp, (Aug. 26, 1994) (noting that the U.S. Department of Justice announced "that it has taken action against more than 150 individuals" accused of phishing and other related spam attacks); *2004: Year of the Cyber-Crime Pandemic, at* http://www.eweek.com/article2/0,1895,1745848,00.asp (Jan. 1, 2005) (noting that between July and Nov. 2004, there was an average monthly growth rate of unique phishing attacks of 34%).

[73] See Lisa Vaas, *No One-Stop Shopping to Stop Database Pilferages, at* http://www.eweek.com/article2/0,1895,1904527,00.asp (Dec, 29, 2005) (describing attack on database of role-playing game company where attackers "exploited a software flaw and threatened to post stolen user data including user names, e-mail addresses and encrypted passwords" unless they were paid).

[74] Bob Keefe, *New Worm is Thief, Not Prankster,* THE ATLANTA JOURNAL CONSTITUTION, Aug. 20, 2005 *at* 1G (detailing how criminals exploited a vulnerability in Microsoft software to "quietly 'harvest' ... sensitive data on a small number of computers – employee Social Security numbers, credit card numbers, passwords" – and then turn the machines into networks of "bots," to be "sold on virtual black markets.").

[75] Gavin Clarke, *Windows beats Linux-Unix on Vulnerabilities – CERT, at* www.theregister.co.uk/2006 /01/05/windows_linux_unix_security_vulnerabilities (Jan. 5, 2006).

[76] Brian Krebs, *Windows Security Flaw is 'Severe,'* WASHINGTON POST, Dec. 30, 2005 *at* D1.

[77] U.S. Government Accountability Office, *Elections: Federal Efforts to Improve Security and Reliability of Electronic Voting Systems Are Under Way, But Key Activities Need to Be Completed,* at 29 (Sept. 2005) (Report No. GAO-05-956) [hereinafter GAO Report] available *at* http://reform.house.gov/UploadedFiles/ GAO-05-956.pdf.

[78] Brendan I. Koerner, *Welcome to the Machine,* HARPER'S MAGAZINE Apr. 1, 2004, at 83.

[79] *Id.; See also* Wikipedia entry for Ron Harris, http://en.wikipedia.org/wiki/Ron_Harris_(programmer) (as of May 30, 2006 15:00 GMT).

[80] In computing, "a patch is a small piece of software designed to update or fix problems with a computer program. This includes fixing bugs, replacing graphics and improving the usability or performance." *See* Wikipedia, *Software Patch,* http://en.wikipedia.org/wiki/Software_patch (as of May 26, 2006 15:42 GMT). Also *see* J. G. Levine *et. al., Detecting and Categorizing Kernel-Level Rootkits to Aid Future Detection,* IEEE SECURITY AND PRIVACY, Jan-Feb 2006, at 24-32.

[81] On a ballot (whether electronic or paper), candidate names are listed numerically with, say, "1" next to Tom Jefferson's name and "2" next to Johnny Adams. In the ballot definition file, programmers define what those numbers mean so when a voter touches a box next to 1 on the screen, the vote gets tallied for Tom Jefferson.

[82] This is not intended to be an exhaustive list.

[83] *GAO Report, supra* note 78 at 33.

[84] "A rootkit is a set of software tools frequently used by a third party (usually an intruder) after gaining access to a computer system. These tools are intended to conceal running processes, files or system data, which help an intruder maintain access to a system without the user's knowledge. Rootkits are known to exist for a variety of operating systems such as Linux, Solaris and versions of Microsoft Windows. A computer with a rootkit on it is called a rooted computer. The word "rootkit" came to public awareness in the 2005 Sony CD Copyright protection controversy, in which SONY BMG music CDs placed a rootkit on Microsoft Windows PCs." Wikipedia, *Root Kit,* http://en.wikipedia.org/wiki/Root_kit (as of May 30, 2006 15:50 GMT).

[85] See Tadayoshi Kohno, Adam Stubbelfield, Aviel Rubin, and Dan S. Wallach, *Analysis of an Electronic Voting System* at 13-14 (Feb. 2004), at http://avirubin.com/vote.pdf (paper for the IEEE Symposium on Security and Privacy); Dr. Michael A. Wertheimer, RABA Technologies LLC, *Trusted Agent Report: Diebold AccuVote-TS System* at 8 *available at* http://www.raba.com/press/TA_Report_AccuVote.pdf (Jan. 2004) (report prepared for Department of Legislative Services, Maryland General Assembly, Annapolis, Md.), [hereinafter "RABA Report"].

[86] *GAO Report, supra* note 78 at 25.

[87] The five points of vulnerability listed here are not meant to be a complete list; rather they represent some of the most obvious points of attack.

[88] *See,* Harri Hursti and Eric Lazarus, *Replaceable Media on Optical Scan,* NIST *at* http://vote.nist.gov/threats/papers/ReplaceableMediaOnOpticalScan.pdf (Last visited May 31, 2006).

[89] Kim Zetter, Diebold Hack Hints at Wider Flaws, WIRED NEWS, Dec. 21, 2005 *available at* http://www.wired.com/news/politics/evote/0,69893-0.html.

[90] *Id.*

[91] "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." *RABA Report, supra* note 86 at 16.

[92] Responses to the Brennan Center Security Survey are on file at the Brennan Center. For sample survey, *see* Appendix D.

[93] *See e.g.* Dean Takahashi, *Cautionary Tales for Security Expert,* PROCESSOR, Mar. 25, 2003 *available at* http://www.processor.com/editorial/article.asp?article=articles%2Fp2712%2F03p12%2F03p12.asp&guid=&searchtype=&WordList=&bJumpTo=True (detailing the reporting of security expert Kevin T. Mitnick, who showed how three hackers successfully obtained an old video-poker machine, took it apart and deciphered its software; this allowed them to steal more than $1 million from Las Vegas casinos).

[94] As a reminder, the ballot definition files are created after a machine and its software have been tested and inspected. The files are sent to local jurisdictions and allow the machine to (a) display the races and candidates in a given election, and (b) record the votes cast.

[95] "Personal digital assistants (PDAs or palmtops) are handheld devices that were originally designed as personal organizers, but became much more versatile over the years. A basic PDA usually includes a date book, address book, task list, memo pad, clock, and calculator software. Many PDAs can now access the Internet via Wi-Fi, cellular or Wide-Area Networks (WANs) or Bluetooth technology. One major advantage of using PDAs is their ability to synchronize data with a PC or home computer." Wikipedia, *Personal Digital Assistant, at* http://en.wikipedia.org/wiki/Personal_digital_assistant (as of May 26, 2006 15:45 GMT)

[96] A Cryptic Knock is an action taken by a user of the machine that will trigger (or silence) the attack behavior. The Cryptic Knock could come in many forms, depending upon the attack program: voting for a write-in candidate, tapping a specific spot on the machine's screen, a communication via wireless network, *etc.*

[97] This is the testing of the tabulator setups of a new election definition to ensure that the content correctly reflects the election being held (*i.e.*, contests, candidates, number to be elected, ballot formats, *etc.*) and that all voting positions can be voted for the maximum number of eligible candidates and that results are accurately tabulated and reported.

[98] For a more detailed discussion of specific attacks, *see* http://vote.nist.gov/threats or request a copy of the *Attack Catalogs* at www.brennancenter.org.

[99] *RABA Report, supra* note 86, at 20-21.

[100] A more complete description of the testing and inspection process for machines (touched upon in pages 42–44), can be found in Appendix E.

[101] By "inspection" we mean review of code, as opposed to "testing," which is an attempt to simulate voting to ensure that the machine is functioning properly (and votes are being recorded accurately). We discuss testing in the next subsection.

[102] David M. Siegel, an independent technology consultant for this report, contributed significantly to this subsection. For a more detailed discussion of the difficulty of catching attack programs through inspection, *see* Ken Thompson, *Reflections on Trusting Trust,* 27 COMMUNICATION OF THE ACM 761 (Aug. 1984), *available at* www.acm.org/classics/sep95.

[103] This is a software program that is generally sold as commercial off the shelf software.

[104] For further discussion of the limits of ITA testing and State Qualification Tests, *see GAO Report, supra* note 78 at 35; Douglas Jones's *"Testing Voting Machines", at* http://www.cs.uiowa.edu/~jones/voting/ testing.shtml#ita (Last visited May 30, 2006); Dan S. Wallach, *Democracy at Risk: The 2004 Election in Ohio, Section VII: Electronic Voting: Accuracy, Accountability and Fraud,* DEMOCRATIC NATIONAL COMMITTEE VOTING RIGHTS INSTITUTE, at 4 (June 2005), *available at* http://www.votetrustusa.org/pdfs/ DNCElectronic%20Voting.pdf

[105] "Firmware is software that is embedded in a hardware device" (*i.e.,* the voting machine). Wikipedia, *Firmware, at* http://en.wikipedia.org/w/index.php?title=Firmware&oldid=48665273 (as of May 26, 2006 15:25 GMT)

[106] Election Assistance Commission, *Voting Systems Standards Volume II, National Testing Guidelines* at §1.3.1.3, *available at* http://www.eac.gov/VVSG%20Volume_II.pdf (Last visited May 30, 2006).

[107] *GAO Report, supra* note 78 at 35-6.

[108] For a complete description of testing that a voting machine might be subject to, refer to Appendix E.

[109] Some voters sign in but never vote (or finish voting). Thus, it might be possible to subtract votes from one candidate without altering the poll books and still prevent the attack from being

noticed. An attacker would be limited, however, in the number of votes she could subtract from a candidate without raising suspicion.

[110] In general, computer systems are programmed to record many activities that occur – including when they are started up, when they are shut down, *etc.* A voting terminal could be programmed to remember when it was started, shutdown, when it printed its zero tape, and the like. Such records are Event Logs or Audit Logs. Ordinarily, these records could be helpful during a forensic analysis of voting systems after a suspected attack.

[111] This presupposes there is no paper record, or that if there is such a record, it is not reviewed.

[112] Acronym for "basic input/output system." The BIOS is the built-in software that resides on a Read Only Memory Chip (ROM) that determines what a computer can do without accessing programs from a disk. Because the software is built-in to the machine, it is not subject to ITA inspection. It could both (a) contain an attack program and (b) delete entries from an Audit Log that might otherwise record the attack.

[113] Independent investigators have already established that this is possible against multiple systems. As noted in the *GAO Report,* "Evaluations [have shown] that, in some cases, other computer programs could access ... cast vote files and alter them without the system recording this action in its audit logs." *GAO Report, supra* note 78 at 25. *See also* Compuware Corporation, *Direct Recording Electronic (DRE) Technical Security Assessment Report* at 42, (Nov. 2003) (prepared for the Ohio Secretary of State), *at* http://www.sos.state.oh.us/sos/hava/compuware112103.pdf; Harri Hursti, *The Black Box Report: SECURITY ALERT, Critical Security Issues with Diebold Optical Scan Design* at 18 (July 2005), at http://www.blackboxvoting.org/BBVreport.pdf; Michael Shamos, *UniLect Corporation PATRIOT Voting System: An Evaluation* at 11 (Apr. 2005) (paper prepared for the Secretary of the Commonwealth of Pennsylvania) *available a*t http://www.house.gov/science/hearings/ets04/jun24/shamos.pdf

[114] Coordinating software attacks with paper records attacks is discussed in greater detail *infra* pp. 68–74.

[115] This assumes an audit of the voter verified paper record is conducted after voting is complete.

[116] It is possible that an attack program could instruct a DRE printer to cancel votes and print false paper records to match attacked electronic records. This points to the importance of examining cancellations on VVPT printouts, as discussed *infra* pp. 69–71.

[117] *See e.g.,* Kim Zetter, *Did e-Vote Firm Patch Election?,* Wired News Oct.13, 2003 (noting that employee of voting machine vendor claimed uncertified software patches were sent to election officials throughout Georgia to install just before the 2002 gubernatorial election) *available at* http://www.wired.com/news/politics/0,1283,60563,00.html; Andrew Orlowski, *California Set to Reject Diebold e-Voting machines* (Apr. 24, 2004) (noting that voting machine vendor sent software updates to voting machines in California just two weeks before the Presidential Primary in that state) *at* http://www.theregister.co.uk/2004/ 04/24/diebold_california.

[118] For a more detailed list of these potential attacks, as well as the steps and informed participant values assigned to them, *see* the "DRE without VVPT Catalog," *Attack Catalogs, supra* note 20

[119] This summary borrows heavily from *"Trojan Horse in DRE -OS"* posted by Chris Lowe for the NIST Threat Analysis Workshop in Oct. 2005. A copy of that posting (which provides a more complete description of the attack) can be found *at* vote.nist.gov/threats/papers/TrojanHorse-DRE-OS.pdf

[120] In fact, this is not a hypothetical scenario. We know that most voting systems run on commercially available operating systems. For instance, at least one major vendor runs its machines on a version of Micrsoft Windows called "CE." It is not difficult to imagine that one of the vendor's

software developers could install such a Trojan Horse without detection.

[121] In this sense, this attack would not require the assistance of an "insider," such as a leading state or county election official.

[122] As already discussed, such updates and patches are issued on a fairly regular basis. For instance, on Jan. 6, 2006, Microsoft issued a patch to address a security flaw found in its operating system. John Fontana, *Microsoft Rushes out Patch for Windows Metafile Attack*, PC WORLD, Jan. 6, 2006 *available at* http://www.pcworld.com/news/article/0,aid,124246,00.asp.

[123] This assumes that the same DRE system is purchased by every county. Obviously, to the extent that the attackers wanted to attack more than one type of DRE system, they might need additional participants in their conspiracy.

[124] As already discussed, *supra* pp. 36–38, there are many ways for an attacker to gain such knowledge.

[125] Appendix G

[126] Of course, few states use a single make and model of machine in every county. But even if a single DRE model represented 1 in 3 of all machines in the state, the attacker would need only target those machines and aim to switch between 4 and 6 votes per machine to affect tens of thousands of votes and change the results of the statewide election.

[127] In any event, even where code is subject to inspection, bad code can still get through. In separate instances in California and Indiana, election officials discovered that uncertified software had run on voting machines during elections. *See Marion County Election Board Minutes(Emergency Meeting)* at 7-18, (April 22, 2004) (Indiana) *available at* http://www.indygov.org/NR/rdonlyres/emkiqfxphochfss2s5anfuxbg j3zgpkv557moi3rb6f3ne44mcni2thdvoywyjcigyeoykwru53mopaa6kt2uxh7ofe/20040422.pdf; Office of the Secretary of State, *Staff Report on the Investigation of Diebold Elections System, Inc.* at 1-2 (Apr. 2004), (California) *at* http://www.ss.ca.gov/elections/ks_dre_papers/diebold_report_april20_final.pdf. In one case, the discovery was made when a vendor employee told a county clerk; in the other, the uncertified software was revealed during a statewide audit of machines. We do not suggest that the software was installed to change the results of elections. Nevertheless, the fact that uncertified software ran on voting machines during elections, in violation of regulations and state law, demonstrates the difficulty of finding undesirable software on voting machines during inspection.

[128] Exactly what should happen when Parallel Testing finds that tested machines are misrecording votes is something that California (the only state to regularly perform parallel tests in the past) has not yet had to deal with. Obviously, merely finding corrupt software on a tested machine without taking further action will do nothing to thwart a software attack. Parallel Testing is much less likely to be an effective countermeasure if jurisdictions do not have in place clear procedures about what steps should be taken when the script and vote totals on a tested machine do not match.

[129] All of whom would have to be "insiders," in the sense that they would have had to have been chosen by the State or consulting group performing the Parallel Testing.

[130] *See* discussion in Appendix H.

[131] *Id*. This assumes that Pennasota uses the same make and model DRE in every precinct.

[132] *See* calculations in Appendix H.

[133] *Id*.

[134] Interview with Jocelyn Whitney, *supra* note 46.

[135] In fact, this is exactly how California has conducted its Parallel Testing; each Parallel Testing team casts 101 votes. *Id*.

[136] This is because to switch 51,891 votes, Trojan Horses will need to be activated on at least 2883 machines.

[137] *See* Appendix G.

[138] We calculate that a minimum of 61 attackers would be needed to subvert Parallel Testing in this way. The attackers could target 606 polling places in the three largest counties. It would be necessary for each attacker to get close enough to only ten polling places to transmit a wireless instruction to trigger the attack.

[139] Another possibility is that the Parallel Testers may always record the same number of votes. In previous elections in California, exactly 101 votes were processed during each Parallel Test. If the Trojan Horse is programmed to wait until the end of the election to switch votes, it could avoid all Parallel Testing by changing votes only where machines record more or less than 101 votes by the end of Election Day. E-mail from Jocelyn Whitney (Jan. 2, 2005) (on file with the Brennan Center).

[140] An alternative solution to the problem of creating a script that mirrors actual voter patterns would be to select volunteers, or "real" voters, to vote on the tested machines. These volunteers would be asked to vote as they normally would: this might create more realistic voting patterns without a script, but it potentially raises other privacy issues. We are not aware of any jurisdiction that currently performs Parallel Testing in this way.

[141] *Supra* note 135.

[142] E-mail from Office of the California Secretary of State to Eric L. Lazarus, Principal Investigator (Feb. 1, 2006) (on file with the Brennan Center).

[143] The Pennasota governor's race was designed to represent a closely contested statewide election. Our analysis shows that if a Trojan Horse were used to change just one vote per DRE, the result of the governor's race could be changed. In the case of such an attack, a successful Parallel Test would "detect" the misrecording of a single vote. Without a videotape of the testing itself, this misrecording could easily be misattributed to human error (*i.e.*, accidental deviation from the script). Even with video evidence, there may be a temptation to "explain away" such a discrepancy.

[144] Our total for the Parallel Testing set of countermeasures depends upon the ability of the attacker to create an Attack Program that can recognize if it is being tested. As already discussed, we believe that creating such an attack program would be technically and financially challenging – or would require the involvement of someone who was involved in or knew of the testing script – and have therefore agreed that it would probably require two additional conspirators. To the extent creating such an attack program is not feasible, the attack would require the subversion of at least 58 testers (who might be considered "insiders") to use a Cryptic Knock to shut off the Trojan Horse; we believe this would be very difficult to accomplish.

[145] For a more detailed list of these potential attacks, as well as the steps and informed participant values assigned to them, *see* the "DRE w/ VVPT Catalog," *Attack Catalogs, supra* note 20.

[146] There are other potential entry points for parameterization: wireless communications and Cryptic Knocks could also contain commands that tell voting machines when and how to attack a ballot.

[147] Barbara Simmons, *Electronic Voting Systems: the Good, the Bad, and the Stupid*, The National Academy of Sciences, Computer Science and Technologies Board, at 7-8, *available at* http://www7.nationalacademies.org/cstb/project_evoting_simons.pdf (last visited May 30, 2006).

[148] This attack is similar in structure to Ron Harris's attacks against computerized poker and other gaming machines (*see supra* at p. 33): an employee with access to vendor software, hardware or firmware, inserts the Trojan Horse, which will not trigger until an accomplice sends commands.

[149] *See* Appendix G. Based upon interviews with election officials in Nevada, we have concluded that DREs w/VVPT can handle slightly fewer voters per hour than DREs w/VVPT. Accordingly we have estimated that Mega, Capitol and Suburbia county would have to have one DRE w/ VVPT for every 120 voters.

[150] *Recounts, supra* note 42 at 4. A few states, such as New Hampshire, have laws that allow for inexpensive, candidate initiative recounts. Attackers might be less inclined to target such states. The effect of these laws was not a subject of the Task Force analysis.

[151] In fact, it would work exactly the same as any Software Attack Program against DREs, except that it would also target the VVPT to ensure that the paper records matched the electronic records.

[152] Ted Selker and Sharon Cohen, *An Active Approach to Voting Verification* at 2 CalTech/MIT Voting Technology Project (May 2005), *at* http://www.vote.caltech.edu/media/documents/wps/vtp_wp28.pdf.

[153] *Id.* at 5.

[154] Given that many voters are likely to assume the mistake was their own, rather than the DRE's, we are skeptical that the number would be this high.

[155] See Appendix G.

[156] *Supra*, note 135.

[157] Telephone interview with Larry Lomax, Registrar of Voters, Clark County, NV (Dec. 12, 2005).

[158] There are 28,828 DREs w/VVPT in Pennasota. Appendix G

[159] As detailed in Appendix A, we believe 606 polling places (in the three largest counties) is the minimum number of polling places the attacker could target and have a reasonable amount of certainty that she could still change the outcome of the election. If the attacker targeted 606 polling places, there would be approximately 22 more paper cancellations in these polling places than would otherwise be expected (13201/606=22).

[160] *See* Appendix L.

[161] If the attackers intercepted 550 convoys, there would still be 56 polling places with mismatching paper and electronic records. That represents roughly 0.2% of all polling places in the state. Under these circumstances, a 2% ARA would still have a 66% chance of catching a mismatch. *See* Appendix K.

[162] This is because our attackers seek to switch 51,891 votes. To avoid suspicion, they have not switched more than 15% of votes on any single DRE w/VVPT, which equals 18 (of 120) votes. 51,891/18=2,883.

[163] For an explanation as to why nearly all of the paper rolls would need to be replaced in order to have a reasonable chance of avoiding detection during audit, *see* Appendix K.

[164] According to the Department of Defense, these seals can cost as little as one or two cents per seal; the Department of Defense estimates that for several models, it would take a knowledgeable and highly trained person at least several minutes to "defeat" each seal and gain access to the ballots. Telephone interview by Eric L. Lazarus with Mike Farrar, Department of Defense Lock Program, December 15, 2005. After defeating the thousands of seals, attackers would have to find a way to replace each one with a seal that looked exactly the same and contained the same unique number as the original.

[165] If the employees assigned to guard the election materials are selected from a large pool of employees on-duty on Election Night, and if this selection process is done in a transparently random process just before the voter verified paper records arrive at the county warehouse, the attacker would need to co-opt almost all of the larger pool to have a reasonable chance of co-opting the employees eventually chosen to guard the materials. This would make their task much more difficult.

[166] *Recounts, supra* note 42 at 5.

[167] With more than 1,000 voters in many polling places, the attackers could easily replace enough votes to ensure that Johnny Adams overcame his loss.

[168] Cal. Elec. Code §19253(b)(2) (2006) provides that the "voter verified paper audit trail shall govern if there is any difference between it and the electronic record during a one-% manual tally or full recount."

[169] *Recounts, supra* note 42 at 5.

[170] 10 ILL. COMP. STAT. 5/24C-15 (2005).

[171] In their 2004 report, *Recommendations of the Brennan Center for Justice & The Leadership Council on Civil Rights for Improving Reliability of Direct Recording Electronic Voting Systems,* (*at* www.brennancenter.org/programs/downloads/ voting_systems_final_recommendations. pdf), the Brennan Center and the Leadership Conference on Civil Rights recommended that jurisdictions hire independent security experts and create independent security oversight panels to implement and oversee security measures. To the extent that jurisdictions have adopted these proposals, these groups could be present during any forensic investigation to increase its transparency.

[172] Where a state determines that electronic records should be given a presumption of authority, the reverse process would be followed: first investigate the electronic records for tampering, then (if necessary) examine the paper records.

[173] This number depends upon whether the ballot definition file is created at the vendor or by individual counties. If the vendor creates the ballot definition file for several counties in the state, the Trojan Horse can be inserted into the ballot definition files of multiple counties from a central location. Where each county created its own ballot definition files, at least three informed participants would be necessary (as we have assumed that a successful attack in Pennasota would target a minimum of three counties, three separate individuals, with access to each county's ballot definition files would be needed).

[174] A full catalog of the attacks against PCOS that have been examined can be found in *Attack Catalogs, supra* note 20

[175] *See supra* notes 88 and 89.

[176] *See supra* note 90.

[177] The central tabulator is most often employed to perform ballot definition, copying of ballot definition to the memory cards (so that voter choice will be recorded accurately) as well as tabulation of voter choice. The central tabulator is a conventional Personal Computer with additional software added. Accordingly, it provides a convenient single point of attack which one can modify all the print drivers from all the PCOS scanners in a single county.

[178] This estimate is based upon a review of 19 contracts executed by counties around the country for purchase of voting machines. Copies of these contracts are on file at the Brennan Center.

[179] *See* Appendix G.

[180] 7% of 693 votes is 49 votes. If the Software Attack Program targeted 800 machines in the three largest counties, it could switch close to 40,000 votes.

[181] *See* Assumptions in Appendix G; this assumes the same make and model PCOS scanner was used throughout the state.

[182] This is true with one important caveat: if the PCOS scanners had wireless components, or were in some other way connected to each other or a central location, additional attackers could circumvent Parallel Testing via a remote control command that triggered or superseded the attack.

[183] *See supra* at pp. 49–50 (Representative "Least Difficult" Attack: Trojan Horse Inserted Into Operating System, DRE Attack No. 4)

[184] Specifically, in the 2004 Presidential Election, Central Count Optical Scans had a residual vote rate of 1.7%, compared to just 0.7% for PCOS. In counties with African-American populations of greater than 30%, the residual vote rate for Central Count was 4.1%, and for PCOS just 0.9%. Lawrence Norden, *et al., "Voting System Usability"* in The Machinery of Democracy (Brennan Center for Justice ed., forthcoming July 2006).

[185] *Id.*

[186] N.Y. Elec. Law § 7-202 (2006); Minn. Stat. Ann. § 206.845 (2005).

[187] Secretary of State for the State of California, *Decertification and Withdrawal of Approval of Certain DRE Voting Systems and Conditional Approval of the Use of Certain DRE Voting System,* at 7 (Apr. 30, 2004) *available at* http://www.ss.ca.gov/elections/ks_dre_papers/decert1.pdf. ("No component of the [DRE] voting system shall include the hardware necessary to permit wireless communications or wireless data transfers to be transmitted or received.")

[188] Among them are ES&S and WinVote. *See,* Jay Wrolstad, *Florida Invests $24m in Wireless Voting Machines,* Mobile Tech Today (Jan. 31, 2002) *at* http://www.wirelessnewsfactor.com/perl/story/ 16104.html; Blake Harris, *A Vote for the Future,* Government Technology Magazine (Aug. 29, 2003) *at* http://www.govtech.net/magazine/story.php?id=61857&issue=8:2003.

[189] *See,* Krebs *supra* note 77 ("A previously unknown flaw in Microsoft's Windows operating system is leaving computer users vulnerable to spyware, viruses and other programs that could overtake their machines. . . .").

[190] Maryland, which does not require voter verified paper records, also performs election day Parallel Testing. The 12 states that perform must conduct audits of their voter verified paper records after every election are: AK, CA, CO, CT, HI, IL, MN, NM, NC, NY, WA, and WV.

[191] The 26 states are: AK, CA, CO, CT, HI, ID, IL, ME, MI, MN, MO, MT, NC, NH, NJ, NM, NV, NY, OH, OR, SD, UT, VT, WA, WI, and WV.

[192] Laws providing for inexpensive-candidate initiated recounts might also add security for voter verified paper. The Task Force did not examine such recounts as a potential countermeasure.

[193] Some DREs and DREs w/VVPT may be designed so that they cannot function unless they are connected to one another. Election officials should discuss this question with voting system vendors.

[194] Two other states, West Virginia and Maine, ban networking of machines without banning wireless components themselves. Banning the *use* of wireless components (even when that involves disabling them), rather than requiring *removal* of these components, still leaves voting systems unnecessarily insecure.

[195] *See, Recommendations of the Brennan Center for Justice and the Leadership Conference on Civil Rights for Improving Reliability of Direct Recording Electronic Voting Systems* (2004), http://www.brennancenter.org/programs/downloads/ voting_systems_final_ recommendations.pdf (recommending that jurisdictions hire independent security experts and create independent security oversight panels to implement and oversee security measures). Independent security experts and oversight panel members should be present during any forensic investigation, to increase its transparency.

[196] When a state determines that electronic records should be given a presumption of authority, the reverse process should be followed: first investigate the electronic records for tampering, then (if necessary) examine the paper records.

[197] As previously discussed, to ensure the robustness of our findings, we have already run our analysis against the results of the 2004 presidential race in Florida, New Mexico and Pennsylvania.

[198] Many of these definitions are supplemented by text in the report and Appendices.

[199] *Recounts, supra* note 42 at 3.

[200] For further discussion of inspection and testing performed on voting machines, *see* Appendix E.

[201] NIST's *Glossary of U.S. Voting Systems, at* http://xw2k.sdct.itl.nist.gov/lynne/votingProj/main.asp (Last visited June 10, 2006)

[202] Wikipedia, *supra* note 65.

[203] National Security Telecommunications and Information Systems Security Committee, NSA *National Information Systems Security (INFOSEC) Glossary,* NSTISSI No. 4009, at 49 (June 5, 1992), *available at* http://www.cultural.com/web/security/infosec.glossary.html

[204] For a detailed discussion of a history of fraud against paper-based systems through ballot stuffing, vote buying and other methods, *see* HARRIS, *supra* note 16.

[205] This Appendix is largely borrowed from Douglas Jones's "Testing Voting Machines," part of his *Voting Machines Web Pages,* which can be found at http://www.cs.uiowa.edu/~jones/voting/testing.shtml (Last visited June 10, 2006). We thank Professor Jones for permission to use this material.

[206] The importance of making sure that observer/participant understand how the random numbers are to be used is amusingly illustrated in the magic special: *Penn & Teller: Off the Deep End* (NBC television broadcast, Nov 13th, 2005). In this program an unsuspecting individual is fooled into thinking that the magicians could figure out in advance what card he or she will select because, no matter what card is selected, the magicians can point to its representation somewhere on the beach. The humorous approach here is that all 52 playing cards were set up in interesting ways on the beach to be revealed. A magician opened his coat for one card, two kids in the water held up their rafts to form a card, a sunbather turned around with a card painted on her back, cards were found inside of a potted plant and coconut, *etc.*

[207] Based on the parameters we have set for our election in Pennasota, this would be enough machines to swing the election between Jefferson and Adams. Going back to the assumptions made in this report: the attacker will not want to create a swing of more than 15% on any machine; there are 125 votes recorded per machine; this means the attacker will not want to switch more than 18.75 votes per machine; if her program attacks 2883 machines, she will switch 54,056 votes, more than the 51,891 "target" votes to switch listed in Appendix G.

[208] Again, this assumes that the same make and model DRE is used in the entire state. For suggestions on how to perform Parallel Testing when there are several models of DRE in use in the state, *see* page 88 in this report.

[209] Illinois law provides an example of how to make forensic investigations transparent: in the event investigations following a discrepancy revealed in an audit of paper records, the State Board of Elections, State's Attorney or other appropriate law enforcement agencies, the county leader of each established political party in the affected county or counties, and qualified civic organizations be given prior written notice of the time and place, and be invited to observe. 10 ILL. COMP. STAT. 5/24C-15

[210] Again, Illinois provides an example of one way to increase the transparency of the investigation: the State Board of Elections, State's Attorney or other appropriate law enforcement agencies, the county leader of each established political party in the affected county or counties, and qualified civic organizations are given prior written notice of the time and place of all forensic investigations of machines or paper, and are invited to observe.

BRENNAN CENTER FOR JUSTICE
BOARD OF DIRECTORS AND OFFICERS

BRENNAN CENTER
FOR JUSTICE
AT NYU SCHOOL OF LAW
161 Avenue of the Americas
12th Floor
New York, NY 10013
212-998-6730

www.brennancenter.org

BRENNAN CENTER
FOR JUSTICE
AT NYU SCHOOL OF LAW
161 Avenue of the Americas
12th Floor
New York, NY 10013
212-998-6730

www.brennancenter.org