Voting Systems Audit Log Study

David Wagner
University of California, Berkeley

June 1, 2010

Contents

Executive Summary

This report documents the findings of a study commissioned by the California Secretary of State to examine voting system audit logs. In this study, I examined the audit logs produced by six voting systems approved for use in the State of California. I identifed a number of criteria for evaluating audit logs, and used voting system documentation to assess the strengths and weaknesses of these audit logs.

First, I assessed the completeness of the systems' audit logs, as best as possible given the documentation I reviewed. I found that all of the voting systems record events in their audit logs under a number of situations, and this information could be useful to auditors in some circumstances. At the same time, I found opportunities for improvement. In particular, there are some kinds of relevant events that are not recorded in audit logs. I found that the degree of coverage varied from voting system to voting system. Based upon the documentation available to me, the Hart voting system appeared to have the most complete audit logs of the voting systems I studied.

Next, I assessed the support that the voting systems provide for collecting, managing, and analyzing these audit logs. I found that only one voting system, the Hart voting system, provides a good way to collect audit log data after each election. The rest of the voting systems provide no support for log collection after each election. I found that none of the voting systems provide tools or other support for analyzing audit logs, generating summary reports, or extracting actionable conclusions from the log data. I found that most of the voting systems do not provide clear and complete instructions for how election officials can archive all audit logs after each election; in some cases, the system does not appear to provide any good way to collect and archive all log data. Each of these limitations impairs the usefulness of the voting system audit logs.

I also assessed the degree to which the voting systems support third-party access to audit logs and the ability of observers, candidates, and members of the public to make sense of the log data. I found that the voting systems generally provide poor support for this use of audit logs. Audit logs are often stored in proprietary file formats that are not documented in any publicly available documentation. Based upon my review of voting system documentation, it appears that most of the voting systems do not provide tools for exporting audit log data to an open format suitable for third-party analysis. These limitations raise barriers to analysis or use of audit log data by election observers, candidates, political parties, members of the public, and others.

In summary, I found that the voting system audit logs have some concrete positive aspects, but I also identified a number of weaknesses and limitations in the logs provided by these six voting systems. There seems to be room for future improvement, especially in the areas of log collection, management, analysis, and publication.

Chapter 1

Introduction

1.1 Purpose of this study

This report documents the findings of a study commissioned by the office of the California Secretary of State. The purpose of the study was to assess voting system audit logs as they apply to public elections in the State of California. This study was commissioned to assist the Secretary of State's investigation of the audit logs produced by six voting systems approved for use in the State of California.

1.2 Scope of this study

This study examines six voting systems approved for use in the State of California: DFM Associates' BCWin Ballot Counting application, Election Systems & Software's Unity, Hart Intercivic's Ballot Now, Los Angeles County's Microcomputer Tally System (MTS) 1.3.1, Premier Election Solutions' GEMS, and Sequoia's WinEDS. The MTS software is the central tabulation component of the Inkavote optical scan voting system. The BCWin software is the central tabulation component of the Mark-A-Vote system.

This study examines which features are necessary to create secure and durable audit logs that are sufficient for diagnostic and forensic audits of an election; to what extent each voting system meets these criteria; and what remedial measures may be available for improving voting system audit logs.

1.3 The process followed

The California Secretary of State's office constructed a questionnaire with a series of questions designed to elicit information about the audit logs in these voting systems. The questionnaire was sent to each of the six vendors. I was provided with these questions and the vendors' responses. In some cases, the California Secretary of State's office sent one or more rounds of follow-up questions in response to the vendors' responses, and I received them as well. I reviewed all of those documents carefully. In addition, I was provided with a list of questions and a scope of work for this study from the California Secretary of State's office.

The study included a survey of reports, voting system documentation, scientific publications, and other materials relating to voting system audit logs to inform its findings. I was provided with software and functional specifications from the Technical Data Packages (TDPs) of four of the six voting systems: the ES&S, Hart, Premier, and Sequoia systems. A TDP is a collection of technical documents, including specifications, manuals, and other material, provided by the vendor to the California Secretary of State. Vendors often request that the TDPs be treated as confidential and proprietary; hence these documents typically are not available to the public. I also received the operator's manuals for several of the voting systems. I carefully reviewed each of these TDP documents as they relate to the subject of this study.

I reviewed scientific papers published in the research literature [3, 5, 6, 7, 17, 22, 27, 29, 32, 33, 34, 36, 38, 39, 42]. I reviewed the provisions of the federal voting systems standards, as they relate to audit logs. I reviewed many independent studies of the voting systems considered in this study, including the California Top-To-Bottom Review, Ohio EVEREST, the Florida SAIT Lab reports, reports from the University of Connecticut VoTeR center, the California Diebold AccuBasic review, and others. I reviewed reports that described the use of audit logs in prior elections, to understand how they have been used in post-election investigations. I reviewed a guide on log management from the National Institute of Standards (NIST) [30]. I reviewed submissions to a recent NIST workshop on common data formats for voting systems [18].

Based upon all of these documents, I analyzed carefully the features and functionality that a voting system should provide to support effective audit logs. I analyzed the extent to which the six voting systems studied here provide those features and functionality, at best as possible given the information available to me. I used my professional judgement and experience in the areas of computing and elections to assess these systems and attempt to identify the most important strengths and weaknesses of each system's audit logs. This report documents the results of my analysis.

The California Secretary of State's office provided strong support for this study. The Secretary of State's office provided full access to many non-public documents, as well as a generous allotment of time to complete the study. Also, the staff were careful to protect my independence; at no time did the Secretary of State's office attempt to influence my findings or the outcome of this study in any way. I am grateful to the California Secretary of State's staff for their assistance and support.

1.4 About the author

David Wagner is Professor of Computer Science at the University of California at Berkeley, with expertise in the areas of computer security and electronic voting. He has published over 100 peer-reviewed papers in the scientific literature and has co-authored two books on encryption and computer security. His research has analyzed and contributed to the security of cellular networks, 802.11 wireless networks, electronic voting systems, and other widely deployed systems.

Wagner is a founding member of ACCURATE, a multi-institution voting research center funded by the National Science Foundation (NSF) to investigate ways in which technology can be used to improve voting systems and the voting process. In 2006, he participated in an independent investigation of a disputed Congressional election in Sarasota County, Florida, and in 2007, he helped lead a comprehensive review commissioned by California Secretary of State Debra Bowen to examine California's e-voting systems. He currently serves as a member of the Election Assistance Commission's Technical Guidance Development Committee, the federal advisory board charged with helping to draft future voting standards. He has published several peer-reviewed scientific papers on election audits and voting system audit logs.

David Wagner does not speak for the University of California, the California Secretary of State, or any other organization.

Chapter 2

Background

2.1 Definitions

A voting system audit log is a record generated by the voting system of events that may be relevant for assessing the performance of the voting system and the election processes used with this voting system. In the voting context, an audit log is typically a list of events that have occurred during the conduct of the election or throughout the lifetime of the voting equipment. Generally speaking, audit logs provide evidence that may be examined in the event of a dispute or investigation. In some cases audit logs may also track the actions taken by individual election workers, as a means of accountability.

Because audit logs are an electronic record of events that occur throughout the election, they are sometimes also known as event logs. The two terms are often used interchangeably. The term "event log" is arguably more appropriate [28], because these logs typically contain a list of events that occurred during the election and the time at which each event occurred, and because these logs are only a small part of auditing an election and are not on their own sufficient to ensure that the election outcome will be auditable. Nonetheless, the term "audit log" appears to be more widely used at this point in time. Therefore, for uniformity, we will use the term "audit log" in this report.

In this study, I focus only on electronic records produced by voting equipment. I do not consider paper records such as voter-verified paper audit trails (VVPATs), "zero tapes" printed on election morning, or "summary tapes" printed at the close of elections, even though those paper records may be useful for audits and post-election investigations. Instead, in this study I focus on electronic audit logs. However, I note that, because all California voting systems are required to produce a voter-verified paper record, all of the voting systems considered in this study are auditable in the sense that it is possible to manually recount the voter-verified paper records and cross-check the electronic tallies.

This study focuses on audit logs. Even though cast vote records may be useful in election audits and investigations, they raise different issues. In particular, electronic cast vote records pose special challenges: they are a record of a voter's vote and thus may introduce special privacy and integrity concerns. For this reason, I treat electronic records of cast votes separately from the audit logs.

Audit logs may be generated by each component of the voting system. Equipment deployed to the polling place, such as precinct-count optical scanners or touchscreen voting machines, might generate log entries as each voter interacts with the machine to cast their vote, and as the machines are operated by poll workers. Central-count optical scanners deployed at county election offices might log information as they are used to scan ballots. Election management software might log what happens as county election workers use the software to administer the election, including tasks such as defining the election contests and candidates, laying out the ballot, programming polling-place equipment, testing the equipment, counting ballots, tabulating votes, performing the official canvass, and reporting election results.

2.2 How audit logs are used

Any analysis of voting system audit logs must consider how these logs will be used. There appear to be at least three major categories of uses of audit log data:

- **Routine post-election assessment.** Election officials could informally examine audit logs after every election to assess the performance of the voting equipment and identify opportunities for future improvement. For instance, election officials could potentially scan audit logs to identify anomalous situations, such as precincts where the voting equipment failed, where polls were opened late or closed early, or where other unexpected events occurred. In this way, audit logs might provide statistical information on the reliability of the voting equipment, or might provide insight into how well the election procedures worked in a particular election. Election officials might also use audit logs to identify precincts where further investigation might be warranted. To my knowledge, these potential uses of audit logs are not widespread today, but they could be a possibility for the future.

  These uses of audit logs would require that audit logs be routinely collected; that the voting system provide tools for collating, analyzing, and summarizing audit logs; and that election officials have some way to quickly obtain a short summary report highlighting the most relevant items from the audit logs. Election officials might be concerned that the audit log system be easy to use and easy to adopt, require minimal training, take little time to use, avoid burdening election workers, and generate actionable information that is likely to be of direct relevance to their day-to-day duties.

- **Targeted investigation of election anomalies.** Audit logs can also be used to investigate specific election anomalies and diagnose their cause. In the event of an election anomaly or a dispute or public controversy over some aspect of the conduct of an election, audit log records may be useful to diagnose the cause, nature, and impact of the election anomaly. In some cases audit logs may provide evidence that is potentially relevant to allegations regarding the reliability of equipment, the proper conduct of poll workers, or other concerns. Thus, investigators who have been tasked with examining a specific aspect of a disputed election may find audit logs useful in their investigation. This kind of targeted investigation might be conducted only in special circumstances where a specific, unusual allegation has been raised, instead of after every election.

  These uses require that investigators have a way to collect all of the audit logs, as part of the investigation. Investigators may be particularly concerned with the completeness and coverage of the audit log data and in their ability to process audit log data using their own tools. Candidates, election observers, and interested members of the public might be concerned with their ability to gain access to audit log data and their ability to make sense of the data.

- **Forensic post-election examination.** In exceptional cases, election officials or the legal process might demand a full-scale forensic audit of the voting system, to search for any sign of fraud, misconduct, system failure, or criminal acts. Full-scale forensic audits involve a thorough, in-depth examination of election records and data, including records and data that would not normally be examined by any person, and they might require the participation of experts in forensics, law enforcement, and election systems. For these reasons, forensic audits are time- and resource-intensive and thus can be expected to be rare events.

  A forensic auditor would likely want audit logs to be as detailed as possible. When it comes to a forensic audit, no event or record is too trivial; any piece of information, no matter how minor, might provide the crucial clue. A forensic auditor might also be concerned about the integrity of the audit logs and their chain of custody.

```
Votronic  PEB#   Type    Date       Time     Event

5140052  161061  SUP   03/07/2006 15:29:03   01 Terminal clear and test
         160980  SUP   03/07/2006 15:31:15   09 Terminal open
                       03/07/2006 15:34:47   13 Print zero tape
                       03/07/2006 15:36:36   13 Print zero tape
         160999  SUP   03/07/2006 15:56:50   20 Normal ballot cast
                       03/07/2006 16:47:12   20 Normal ballot cast
                       03/07/2006 18:07:29   20 Normal ballot cast
                       03/07/2006 18:17:03   20 Normal ballot cast
                       03/07/2006 18:37:24   22 Super ballot cancel
                       03/07/2006 18:41:18   20 Normal ballot cast
                       03/07/2006 18:46:23   20 Normal ballot cast
         160980  SUP   03/07/2006 19:07:14   10 Terminal close
```

Figure 2.1: An example of an audit log produced by an ES&S iVotronic machine, reproduced from [38]. We can see that the polls were opened on this voting machine at 3:31pm on election day, zero tapes were printed, a number of ballots were cast, and then the polls were closed on this voting machine at 7:07pm. The iVotronic is not used in California, but its audit log contains information that might be logged by many other voting systems as well.

There are many parties who may have an interest in obtaining access to audit log data, including election officials and other election workers, election observers and other interested members of the public, candidates, political parties, and their representatives, newspapers and other media, and the legal system, including criminal investigators, prosecutors, election lawyers, and judges. In addition, it is possible that developers and engineers who work for the voting system vendor might find audit logs of assistance in tracking down reported problems with the voting system and providing product support to the users of the system. The goals of these different parties may be different. An audit log system must take all of these purposes and uses into account.

2.3 Examples of audit logs

The best way to conceptualize what is contained in the audit logs produced by existing voting systems is probably to look at an example of a voting system audit log. See Figure 2.1 for one example.

2.4 Voting system standards

There have been a series of federal voting system standards. Each one contains minimum requirements regarding voting system audit logs.

1990 standards. The 1990 FEC voting system standards list several requirements on the kinds of audit records that must be generated and how they must be maintained [15, §4.8]. In prefatory remarks, they describe an audit log as a "concrete, indestructible archival record of all system activity related to the vote tally" (§4.8).

2002 standards. The 2002 FEC voting system standards [19] incorporate certain refinements to the audit log provisions of the 1990 standards. Sections 2.2.4.1(g)–(i) of the 2002 standards require systems to

> g. Record and report the date and time of normal and abnormal events;
> h. Maintain a permanent record of all original audit data that cannot be modified or overridden but may be augmented by designated authorized officials in order to adjust for errors or omissions (e.g. during the canvassing process.)
> i. Detect and record every event, including the occurrence of an error condition that the system cannot overcome, and time-dependent or programmed events that occur without the intervention of the voter or a polling place operator

Sections 2.2.5.2.1(a)–(c) require systems to

> provide the capability to create and maintain a real-time audit record

and to timestamp every log entry. Section 2.2.5.3 requires that voting systems that contain COTS (Commercial Off-the-Shelf) operating systems must configure the operating system to log

> all session openings and closings, [...] all connection openings and closings, [...] all process executions and terminations, and [...] the alteration or deletion of any memory or file object.

Section 4.4 specifies certain events that must be logged by any voting system, and also requires vendors to supplement this list with information appropriate to their systems.

VVSG 1.0 (2005 standards). The U.S. Election Assistance Commission (EAC)'s 2005 Voluntary Voting System Guidelines (2005 VVSG), also known as the VVSG 1.0, retain the language found in the 2002 standards, with no significant changes [12].

EAC clarification on audit logs. The EAC recently issued a clarification to the 2002 standards and the VVSG 1.0 (2005 standards), regarding what events must be logged in voting system audit logs [13]. The clarification requires logging of

> any occurrence that may have, alone or in combination with other occurrences, a significant impact upon election data, the management or integrity of the voting system, or configuration, setup, and delivery of the voting and tabulation functions of the system.

It also provides a number of examples of types of events that must be logged.

Proposed VVSG 1.1. The EAC recently released proposed draft revisions to the VVSG 1.0. This proposed draft version is known as the VVSG 1.1 [14]. I have not conducted a careful analysis of the proposed VVSG 1.1, but it appears to retain the provisions regarding audit logs from the VVSG 1.0 with minor changes and clarifications. It also adds several additional requirements that may be relevant. For instance, Section 2.4.4.1 specifies:

> The voting system shall provide the capability to export electronic reports to files formatted in a non-restrictive, publicly-available format. Manufacturers shall provide a specification describing how they have implemented the format with respect to the manufacturers specific voting devices and data, including such items as descriptions of elements, attributes, constraints, extensions, syntax and semantics of the format, and definitions for data fields and schemas.

Section 2.4.4 clarifies that "event logs" are one type of report needed. Section 2.4.4.2 also requires that DREs must be able to export a record of all ballot images. It is worth emphasizing that the proposed VVSG 1.1 is a proposed draft that has not been approved by the EAC; the EAC recently closed a period of public comments on the proposed VVSG 1.1 and at the time of writing is evaluating those public comments.

TGDC's Recommended VVSG 2.0. In 2007, the Technical Guidelines Development Committee (TGDC) delivered to the EAC a proposed draft for next-generation voting standards, known as the VVSG 2.0 [16]. The TGDC is an advisory committee chartered to work with the EAC and NIST to develop voting system standards. The author is a member of the TGDC. The TGDC's recommended VVSG 2.0 was designed to be a ground-up re-think of the federal voting system standards. They are intended to provide guidance towards the systems of the future, not necessarily today's systems.

The VVSG 2.0 has not been approved by the EAC and has not taken effect. Nonetheless, in my opinion it is a useful informational resource that reflects years of effort by the TGDC and NIST. Section 5.7 of the VVSG 2.0 contains detailed, carefully thought-out requirements for audit logs. I would recommend that designers of future voting systems who are concerned with audit logs familiarize themselves with that portion of the VVSG 2.0. Similarly, I would recommend this portion of the VVSG 2.0 to voting system regulators interested in the requirements that a voting system audit log should satisfy.

2.5 Past use of audit logs

It is instructive to examine how audit logs have been used in past elections, to better understand the role they play in elections and the needs they must meet.

Alameda County, California, November 2004. The November 2004 general election in Alameda County included a ballot measure, Measure R, that failed in a very close election. Measure R supporters requested election records, including voting system audit logs, from Alameda County. When the county denied those requests, the supporters filed a lawsuit demanding access. During litigation, the county argued that release of audit logs from the county's central election management system would pose a threat to the security of elections, expressing a "grave concern" that disclosure of "variable names" found in the GEMS audit logs might enable malicious individuals to "hack" future elections [35]. The County's chief election official expressed his view that, in light of his duty to act in the public interest, he felt obligated to withhold access to the audit logs, to protect future elections. In comparison, expert witnesses Doug Jones and Matt Bishop testified that these audit logs do not reveal any information that would enable an attacker to hack a future election. In 2007, the court ruled that the audit logs and other materials must be released to voters who request them and ordered a new election for Measure R. The court also found that the county had not met its obligation to preserve audit logs and other data contained on the county's voting machines.

This experience suggests that the obligation to preserve election records for 22 months may include a duty to archive audit logs, and therefore that voting systems must provide the capability to do so. It also suggests that concerns and uncertainty about the security implications of releasing audit log data have the potential to impede public oversight of elections and diminish transparency [29].

Webb County, Texas, March 2006. In the March 2006 primary election in Webb County, Texas, a judicial race between incumbent Manuel Flores and challenger Joe Lopez was extremely close: a margin of victory of about 100 votes, out of about 50,000 votes cast. Lopez hired expert witnesses to examine the audit logs and check the validity of the results. Lopez's expert witnesses examined audit logs recorded by the ES&S iVotronic voting system, and concluded that (based upon their analysis of the logs) 26 test votes had been inappropriately counted and included in the certified election tally. They also reported that several iVotronic voting machines had been inappropriately cleared in the middle of election day, potentially causing the unrecoverable loss of an unknown number of votes [42, 38]. In the end, Lopez conceded to Flores [43].

This examination demonstrated that audit logs can provide additional insight and evidence into the conduct of the election and the extent to which proper procedures were followed. It also raises the possibility

that, in some cases, audit log analysis might be able to provide some level of comfort to a losing candidate that the election was fair and the outcome legitimate.

Lopez's expert witnesses did mention several challenges that they encountered in their analysis of the Webb County audit logs. They found that the timestamps in the audit logs occasionally appeared to be anomalous, and it was often difficult to determine whether the clocks on those machines were simply incorrect or whether the anomalous timestamp represented a more serious procedural failure. For instance, they found 30 machines where the audit logs showed votes being recorded at least one day before or after election day. Subsequent analysis suggested that for 4 of these machines, the voting machine's clock was probably simply incorrectly set, but it is likely that the other 26 cases represented a procedural failure that caused test votes to be improperly counted. Their experience suggests that the accuracy and veracity of audit log timestamps may pose a challenge for future analysis of audit logs.

Sarasota County, Florida, November 2006. In 2006, Florida's Congressional District 13 was closely contested. Immediately after the election, observers discovered highly anomalous results in Sarasota County, where no vote was recorded in the CD13 contest for 14% of the voters who voted in Sarasota on the ES&S iVotronic electronic voting machine. Because the race was so close, the number of undervotes greatly exceeded the margin of victory, triggering public scrutiny upon the cause of the anomalous undervote. In light of this surprising anomaly, the State of Florida conducted an investigation of the Sarasota County undervote. In a study commissioned by the State of Florida Division of Elections, seven other computer scientists and I worked together to understand the nature, cause, and consequences of the anomaly. As part of our investigation, we had access to audit logs and ballot images (cast vote records) from all of the iVotronic machines in Sarasota County.

One of the questions we investigated was whether the voting machine had properly displayed the CD13 contest and stored voters votes. A number of voters complained that they had never been given a chance to vote in the CD13 contest, or that the voting machine changed their selection after it was made but before the summary screen was reached. Unfortunately, we quickly discovered that the audit logs were too limited to enable us to check these voter reports. The audit logs recorded the fact that a ballot was (or was not) cast, but not the sequence of contests or screens presented to the voter, the selections made by the voter, or the contents of the summary screen when it was displayed to the voter. As a result, it was not possible to reconstruct from the audit logs what the voters saw and did on the iVotronic voting machines. Ultimately, our analysis of the audit logs did not identify the cause of the Sarasota anomaly, though it did reveal two minor software bugs whose impact were relatively benign and could not have caused the anomaly.

One lesson I learned is that while audit logs do have value, the audit logs produced by today's voting systems may be limited in their utility for investigating unforeseen election anomalies, because the logs record so little information about what happened during the election. Their utility is particularly limited where human factors considerations may be a contributing factor, because today's election systems do not log enough information to reconstruct details of the voter's interaction with the machine. As a result of this experience, several other researchers and I have worked to develop novel audit log technology designed to eliminate these limitations of existing audit logs [17, 18, 33, 36].

State of Connecticut, November 2008. After the November 2008 general election in Connecticut, researchers at the University of Connecticut worked in collaboration with the Connecticut Secretary of State to conduct a ground-breaking project to collect and analyze the audit logs from a large number of Connecticut voting machines [3]. Connecticut used Premier AV-OS optical scanners throughout the state in that election. The researchers found that there was no documentation available to them on the format of the audit logs produced by these scanners, and no tools provided by the vendor for efficiently collecting and analyzing these audit logs. To deal with these challenges, the researchers built their own tool for downloading audit

logs from the AV-OS, reverse-engineered the AV-OS audit log file format, and implemented custom tools for analyzing these audit logs. The researchers also developed novel techniques for analyzing the audit logs to identify anomalous events, events that "should be impossible", and other potentially useful information. The researchers used their custom tools to obtain and analyze audit logs from 421 AV-OS machines—fully 30% of the scanners used in Connecticut's November 2008 general election.

The researchers' analysis was able to identify interesting and useful information about the conduct of the election. For instance, they identified 15 machines that were repeatedly restarted, and 10 where counting was restarted in the middle of election day, likely indicating precincts that experienced difficulties with the scanners. As another example, they identified 29 machines that were opened prematurely and 1 machine that was not closed until 10pm at night (possibly due to long voter lines). They also identified several kinds of violations of Connecticut procedure, including 4 violations that could have caused the loss of vote data had they occurred on election day, suggesting that audit logs may be able to provide a useful window into the effectiveness of poll worker training at helping poll workers comply with official procedures.

This experience suggests that routine post-election analysis of audit logs could potentially provide election administrators with useful insights and actionable information about the experience of their poll workers. It also shows the value of tools to collect and manage audit logs. It suggests that the lack of such tools, as well as the lack of public documentation on the format of such tools, can potentially pose a hurdle or barrier to beneficial uses of audit logs. On the other hand, it also suggests that third parties may be able to surmount these hurdles in some cases, given sufficient effort and resources, and that third parties may be able to deliver useful tools that cannot be obtained from any voting system vendor.

Humboldt County, California, November 2008. In November 2008, Humboldt County used Premier's GEMS 1.18.19 system together with Premier AV-OS central count optical scanners to scan, count, and tabulate paper ballots cast in the November election. At the same time, the County engaged in an innovative pilot project, called the Humboldt County Election Transparency Project, to audit the election results by scanning the paper ballots using a separate document scanner and re-tabulating the ballots using an independent open-source software system built by an Election Transparency Project volunteer. As a result of this pilot project, Humboldt County discovered that the GEMS system had omitted 197 valid ballots from the official certified election results. A subsequent investigation determined that this failure was a result of a serious error in the GEMS software used in Humboldt County. Under certain common conditions, the first deck of paper ballots scanned would be deleted at some point after they were scanned into GEMS [11].

Further investigation revealed that the GEMS 1.18.19 software contains additional software flaws which exacerbated this issue. These flaws relate to the audit logs produced by GEMS. In particular, the investigation found [11, §II]:

> First, GEMS version 1.18.19 fails to record in any log important system events such as the deletion of decks of optical scan ballots after they have been scanned and entered into the GEMS election results database. Second, it records the wrong entry date and time for certain decks of ballots. Third, it permits deletion of certain audit logs that contain — or should contain — records that would be essential to reconstruct operator actions during the vote tallying process.

The report also revealed that GEMS 1.18.19 contains a usability design flaw that introduces the risk of accidental deletion of audit logs: on the screen used to view, print, or save audit logs, there is a button to delete the logs that is displayed next to the buttons to save the logs to a file or close the window. This introduces the risk that a minor mouse mis-click could lead to unwanted and unintentional deletion of important audit logs.

GEMS 1.18.19 is not the most recent version of the Premier GEMS voting system, and the Secretary of State has since banned its use in California. In a public hearing conducted by the California Secretary

of State's office after their investigation, a representative from Premier provided further information about the status of these flaws in more recent versions of GEMS [10, pp.24–27]. The representative clarified that subsequent versions of the Premier GEMS system address the third audit log issue identified above (ability to delete logs), by removing the button to clear the audit logs from the user interface. The representative explained that the second audit log issue (incorrect timestamps) is related to the deck zero issue, though it is not clear whether this issue is fixed in subsequent versions of GEMS. The representative stated that the first audit log issue (failure to log deck deletion events) remains present in all versions of GEMS and had not been fixed at the time of the public hearing.

Wired News reported on the issue and cited several concerns with the design of the GEMS audit logs raised by technical experts [44]:

- The logs are "cryptic and obscure", which "destroys [their] value in terms of election transparency", according to University of Iowa professor Doug Jones.
- The system splits log entries across multiple files that do not share a common file format. For instance, the log files do not use the same format for storing timestamps, even though timestamps are one of the most fundamental fields in a log file. The system does not provide any support for collating these log files, collecting them into a shared format, or analyzing them.
- The format of the log files is not documented in any public document, making it difficult at best to understand and interpret the log files. Jones explained, "From the point of view of actually doing any forensics, it's a mess. Because you have to understand what all of the logs are saying, and all of the documentation to understand what they're saying are not public documents."

In my view, the Humboldt County situation provides several thought-provoking lessons about the design of audit logs. It illustrates the importance of good user interface design, to prevent accidental deletion of audit logs. It suggests that we may want to re-think whether voting systems should support deleting audit logs at all. It provides motivation for ensuring that audit logs cover all relevant system events, so that it is possible to detect and diagnose problems like these more effectively. It indicates that it might be useful to have a way to collect all log files into a single place and to analyze the resulting collection of log files. It suggests that, to maximize their utility, log file formats should be publicly documented. And it suggests that the federal certification process is not sufficient to ensure that voting system audit logs are complete and accurate.

Chapter 3

Criteria for effective audit logs

3.1 Answers to specific questions

This section attempts to answer a series of questions posed by the California Secretary of State's office, based upon the document review conducted during this study.

3.1.1 What events should be logged?

Answer. There is no single right answer to this question. Generally speaking, the more that is logged, the more useful that audit logs will be to election investigators—it is sometimes not easy to anticipate what information might be useful, so it is better to have too much data than too little. However, there is a cost to increasing the amount of logging, in the amount of development effort required, in the amount of storage required, and potentially in privacy implications.

At a bare minimum, voting system components should log the events required by the federal standards. However, the federal standards specify only a bare minimum; logging just the events required by the federal standards and nothing more will leave forensic and diagnostic auditors without important information that could help them. For this reason, I identify below some examples of additional information that might be useful to log. The type of events that should be logged by a component of the voting system depend upon the type of component: e.g., central election management system, polling-place vote-capture device (DRE, precinct optical scanner), voter card activator.

Every event that is logged should be accompanied by a timestamp indicating the time at which the event occurred and an indication of which device generated the log entry.

Events that should be logged by every component. Every component should log an event when:

- The component is turned on.
- The component is turned off.
- The component's software or firmware is upgraded or modified. In this case, the component should store the version number or other identifier identifying the previous and new software/firmware. It would be helpful to also store a secure cryptographic hash of the previous software/firmware and of the new software/firmware.
- Any serious or fatal error occurs. If this occurs, the component should also log as much information about the error as possible.

- Any unhandled exception or error occurs (e.g., an exception is thrown, is not caught by internal modules, and propagates up to "the top level" of the software). If this occurs, the component should also log as much information about the exception or error as possible.
- A system operator authenticates himself/herself. In this case, the component should log as much is known about the identity of the operator as possible (e.g., poll worker, election official, username) or the method of authentication used (e.g., which level of PIN or password was entered).
- A checksum or cryptographic signature/MAC/hash is incorrect or corrupted.
- The component's clock or internal time is adjusted. The audit log should record both the previous time (before adjustment) and the new time (after adjustment).
- The ballot definition, election definition, or election configuration are changed. It would be helpful to store a secure cryptographic hash of the new definition/configuration.
- If the audit log is reset or cleared, after clearing the audit log an event should be logged indicating the time at which the audit log was cleared and who cleared the audit log.

It would be helpful if any time the component interacts with another component via a real-time communication channel (e.g., a wired or wireless network), the component logged both its own time and the other component's time. This would assist with synchronizing the clocks of the two components and reconciling discrepant timestamps.

It would be helpful if all components also logged an event when:

- System configuration settings are changed. The component should also log the nature of the change (e.g., the prior and new values of the configuration parameter).
- The component's network connectivity status changes (e.g., it is connected or disconnected from a network), for components with network connectivity. The component should also log information about the network it is connected to, when connecting to a network. When the component is turned on, if it supports network connectivity, its network connectivity status at boot time should be logged.
- A removable storage device or other pluggable device is connected or disconnected to the component. The component should also log information about this removable device and which port it was connected to. When the component is turned on, if it supports removable devices, the status of all removable/pluggable devices attached at boot time should be logged.
- A filesystem is mounted or unmounted.

Events that should be logged by every battery-powered component. Every battery-powered component should log the battery status (the amount of charge remaining in the battery) when it is turned on and when it is turned off. It should also log a low-battery event when the battery is nearing depletion, along with the battery status at that point. If the component is forced to power itself off because the battery has run out of charge, it should log that fact as well.

Events that should be logged by polling-place vote-capture devices. Vote-capture devices intended for use at a polling place, such as DREs and precinct-count optical scanners, should log an event when:

- The machine is opened for voting. The values of the public and protective counters should be logged at this time as well.

- The machine is closed for voting. The values of the public and protective counters should be logged at this time as well.
- A ballot is cast by the voter.
- A ballot is cast by the poll worker after the voter leaves without casting their ballot.
- A voter's session with the voting machine is cancelled by the poll worker, without any ballot being cast.
- A voter's session with the voting machine is cancelled by the voter, without any ballot being cast.
- The printer experiences a paper jam, out-of-paper condition, or other error. The log entry should specify what type of error occurred.
- The votes stored on the machine are cleared. In this case, the audit log should record the values of the public and protective counters before the votes were cleared.

Ideally, these machines would also log an event when:

- A zero tape is printed. It would be helpful if the machine logged, in electronic form, an exact copy of everything printed on the zero tape.
- A summary tape is printed. It would be helpful if the machine logged, in electronic form, an exact copy of everything printed on the summary tape.
- Any action is taken/requested by the poll worker. It would be helpful to log the action requested by the poll worker and whether it was completed successfully or not.
- The machine is activated for the next voter, e.g., when a valid voter card is inserted or when the poll worker activates the voting machine. In voting machines that can be activated in multiple ways, it would be helpful if the log entry indicated the method of activation. (For instance, the Sequoia AVC Edge can be activated using either a voter card or by a poll worker override using the yellow button at the back of the machine. It would be helpful to log which method was used to activate the machine.)
- The voter begins interacting with the voting machine. Logging the time at which the voter begins interacting with the machine would help identify the times at which the voting machine is idle. We can assume that the voting machine is in use by a voter from the time when the voter begins interacting with the machine to the time when they cast their ballot, and we can reasonably assume that the machine is idle and not in use by anyone during the period of time from when a ballot is cast to when the machine is activated for the next user. This kind of idle time information would be helpful in measuring the degree of utilization of each machine, which might help refine the allocation of machines to polling places in future elections.
- If the votes stored on the machine are cleared, it might be helpful to record the timestamp of the last vote previously cast on this machine. If a single operation causes both the votes to be cleared and the audit log to be erased, then this information should be recorded to the new audit log (after the old one is erased).
- The poll worker presses a key or button. It would be helpful to log exactly which key or button was pressed. Exception: If the voting system relies upon poll workers to enter a secret PIN via keys or buttons, the PIN should not be logged.

- Possibly, each time a voter changes one of their selections from a summary screen or immediately after seeing a summary screen. This only applies to DREs and electronic ballot marking devices with a summary screen. To preserve voter privacy, the audit log should not record the previous or new selection, but merely the fact that a change was made (and possibly in which contest the voter changed their selection)[1]. This information might be helpful for detecting or diagnosing some kinds of human factors issues.

- For DREs with VVPAT, when voting is closed it might be helpful to log a count of the total number of VVPAT records that were spoiled during that election, and the number of voters who spoiled their VVPAT record at least once. This information is available from the printed VVPAT roll, but it might be helpful to have this information in electronic form in the audit log. This information might be helpful for detecting or diagnosing some kinds of human factors issues as well as for detecting some ways in which malicious software might try to attack an election.

Optical-scan machines should also log an event when:

- A "special" card is entered, such a header card or ender card. The identity of the card and all information on the card should be logged as well.

- The scanner "kicks back" a ballot to the voter (e.g., an overvoted ballot, a blank ballot, a ballot with marginal marks).

- The poll worker or voter requests an override, to accept a ballot that was "kicked back" to the voter and otherwise would have been rejected.

Ballot-marking devices should also log an event when:

- A blank ballot is inserted.

- The ballot is printed and ejected to the voter.

Components that have a touchscreen should also log an event when:

- The screen is calibrated or re-calibrated. In this case, it would be helpful to log the previous calibration parameters and the new calibration parameters.

Events that should be logged by central election management components. Election management components (which typically take the form of election software that runs on central PCs) should log an event when:

- A user attempts to log on. In this case, the identity (username) of the user should be logged as well as whether the attempt was successful.

- A user logs off.

- A report is generated, saved, or viewed. In this case, the component should log which report was generated, saved, or viewed, and how it was accessed.

[1] If this information is logged, it will require special treatment. This information could potentially enable voters to identify their ballot and prove how they voted to anyone else with access to the audit log, so it would be prudent to log this information separately and avoid disclosing it to unknown individuals, or to develop a tool to redact this information from the audit logs before publicly disclosure. See the discussion of Privacy II in Section 3.1.3.

- A vote tally is manually adjusted or changed. In this case, the component should log both the old and new tallies.

- A vote record is changed or overwritten.

Each such log entry should also identify the user who performed or requested that action.

At some point before the election, the election management component should save an archival copy of all ballot styles, the list of contests and choices for each contest on each ballot style, and a sample ballot for each ballot style. At some point after the election, the election management component should save an archival copy of all cast vote records (also known as ballot images) and all vote tallies received from vote-capture devices.

It would be helpful to also log an event when:

- Access controls or privilege for one user are changed. In this case, the component should log the user authorizing the change, the user whose access controls were changed, and what changes were made (e.g., the previous settings and new settings of each part that was changed).

- Any change is made to the database. It would be helpful to log the change that was made (e.g., the SQL UPDATE statement that was executed). Exception: When passwords or cryptographic keys are changed, the system should not log the password or cryptographic key.

There may be other events that should be logged by election management systems. I have not been able to find much information about experience with election management audit logs. The only audit logs I have seen in my own experience have been generated by polling-place components, not by election management components, so my understanding of what events election management components should log is likely to be incomplete.

Events that should be logged by other voting system components. Voter card activators should log an event each time they activate a voter card, along with the serial number of that card and the previous status of that card (activated, disabled).

3.1.2 Are there events that should not be logged for security reasons?

Answer. Yes. Voting systems should ensure that the information contained in audit logs does not pose a risk to election integrity. Obviously, we do not want the audit logs to reveal secrets that would make it easy for attackers to hack future elections. Consequently, any secret information whose disclosure would pose a clear threat to election security should not be saved in audit logs.

I am aware of only one category of information that should not be logged, because of security reasons:

- **Cryptographic secret keys, passwords, PINs, etc. should not be logged.** In many systems, there is some information where the security of the system relies upon the secrecy of that information. For instance, if the system uses secret-key cryptography, the security of the cryptography relies upon the secrecy of the secret keys. Similarly, the security of user authentication often relies upon keeping a password or PIN secret. For this reason, cryptographic secret keys, secret passwords, and secret PINs should not be logged to audit logs. In general, if the security of the system relies upon the secrecy of some piece of information, that information should not be saved in audit logs. This kind of information is rarely relevant to an audit in any case, so there is no reason to include it in the audit logs and every reason to exclude it.

  It is not acceptable to include this information in audit logs. Even if we somehow knew that audit logs would never be disclosed to the public, it still would not be safe to store this information in audit

logs. There are many parties (poll workers, election staff, county system administrators) who might have an opportunity to access to audit logs, so a well-designed voting system should avoid storing in audit logs any information whose disclosure could endanger the integrity of the election as a whole. Therefore, if there is any credible allegation that audit logs must not be disclosed to the public because the security of the election relies upon the secrecy of certain elements in the audit logs, voting system certifiers should immediately conduct an investigation to re-assess whether the voting system is safe for use in public elections, in light of this information.

Based on my assessment, I have formed the opinion that the best criteria to use is as follows: if the integrity of the election outcome relies upon our ability to keep secret a certain piece of information, then this information must not be included in the audit logs. Other categories of information do not need to be categorically excluded or redacted from audit logs on grounds of election security.

There are other categories of information that might initially raise security concerns, but based upon my assessment I have come to the conclusion that these categories of information are unlikely to require special treatment from voting system regulators:

- **Revealing information about the design or implementation of the system is generally not a serious concern.** If audit log entries reveal information about the design or implementation of the voting system, should this information be excluded from audit logs? Or, if this information is included in audit logs, should election officials treat the audit logs confidentially and avoid disclosing them? How concerned should we be about the security risks of revealing this kind of information? Based on my review, I have formed the opinion that this kind of information is not a serious concern. It generally does not need to be excluded from audit logs, is unlikely to require special treatment, and is unlikely to warrant withholding access to audit logs.

  Sometimes those new to the field of computer security assume that the most effective way to protect computer systems is to assiduously conceal how the system works, reasoning that this has got to make the life of would-be attackers harder. On the surface, this approach feels intuitively reasonable. However, the history of this approach is checquered at best, and at worst it is full of flawed systems that proved much less secure than their designers expected. The approach is so tempting—and so notoriously prone to failure—that it has gained a name among computer security practitioners, who call it "security through obscurity." What we have learned over time is that it is very difficult to prevent attackers from learning how the system works. There are too many ways for attackers to learn this information, whether by reverse-engineering a stolen device, gaining privileged access to information as an insider, learning the information from an insider using social engineering, benefitting from inadvertent leaks, or any number of other means. Moreover, if information about how the system works leaks even once, there is no way to put the genie back in the bottle. Therefore, most computer scientists today consider "security through obscurity" a flawed foundation for systems security, and generally recommend against basing the security of one's system primarily upon concealing how the system works.

  As a result, in any well-designed voting system, revealing information about the design or implementation of the system will not put the integrity of the election at risk. If there is a credible allegation that audit logs reveal specific information about the design or implementation of the system that would endanger the integrity of the election, then this allegation is effectively also alleging that the system is so poorly-designed and its security is so fragile that its security relies upon keeping secret a type of information that cannot reliably be kept secret.

- **Information that might incrementally assist attackers is a grey area, but it is rarely found in audit logs.** There is an intermediate category of information, where the security of the system does

not rely upon the secrecy of this information, but where disclosure of that information might assist attackers to some modest extent.

For instance, in a jurisdiction where election results are uploaded electronically at the close of polls over a modem line, the phone number of the county's central modem bank would be an example of information in this category. In this hypothetical example, the security of the voting system does not rely upon the secrecy of this information, as the modem is only used to transmit unofficial results and as the modem bank will authenticate callers in any case. Indeed, election security had better not depend upon keeping this phone number secret, because there is no way to keep it secret from a dedicated attacker: a serious attacker could use what is known as "war dialing" to automate the process of trying every phone number in the area and in this way identify the modem bank number. Nonetheless, disclosure of the modem bank phone number might reduce the level of effort for such an attacker slightly, so disclosure of the phone number might increase the level of risk by a small amount. Similarly, in jurisdictions that use wireless networks to transmit unofficial election results back to county headquarters on election night, the IP address of the county server would be another example of a piece of information whose disclosure might incrementally increase the risk. In such cases, it might be best to avoid recording this information in the audit logs, simply because there is probably no compelling reason why auditors would need this information. However, as the risk exposure is low, if this kind of information is included in audit logs produced by a legacy voting system, the downside of releasing the information is modest.

None of the examples listed above apply to the State of California, because California does not use modems or wireless networks in polling places. I was not able to think of any other example of information in this category that might plausibly appear in California audit logs. Therefore, I do not believe this category requires special consideration from voting system regulators, and I do not believe it warrants special restrictions on public disclosure of audit logs.

3.1.3 Are there events that should not be logged for privacy reasons?

Answer. Yes. One example is that audit logs should not store the actual votes cast by the voter together with a timestamp indicating the time at which those votes were cast, because this endangers ballot secrecy: it might enable anyone with access to the audit log to identify that voter and determine how he/she voted. Voting systems should be designed to avoid storing in audit logs any information that could reveal the votes of unwitting voters.

There are two fundamentally different voter privacy requirements, which I will call Privacy I and Privacy II for lack of a better name.

- **Privacy I.** A voting system should avoid leaking how a voter has voted, against his/her will. For instance, the voting system should avoid creating any records that link the voter's identity to the votes cast by the voter.

- **Privacy II.** A voting system should protect voters from coercion and protect against vote-selling by preventing voters from proving how they voted, even if the voter wants to. For instance, the voting system should prevent the voters from marking their ballot in a way that would enable them to demonstrate to others which ballot was theirs.

The fundamental difference between these two requirements has to do with whether the voter is trying to reveal how he/she voted. Privacy I relates to helping voters keep their votes secret, while Privacy II relates to preventing voters from proving how they voted even if the voter cooperates with a vote-buyer or coercer.

These two requirements require different protection measures:

- **Protecting Privacy I.** To protect Privacy I, voting audit logs should not save any information that could link a voter's identity to the votes he/she cast.

  Electronic voting systems typically implement this principle by segregating electronic records into two groups. They record information about the votes cast (e.g., cast vote records, ballot images, vote tallies) in one place, while information that might be linked to voter identity (such as the time that a vote was cast) is stored separately, in a way that cannot be linked to the particular votes cast. In particular, cast vote records list the votes that were cast, but contain no information about who cast them: they are not timestamped, and the order in which they are stored is randomized to ensure that the order of stored votes is independent of the order in which they were cast. Audit logs contain information that may be linkable to voter identity, such as the time at which ballots were cast, but are constructed to avoid recording any information about how the voter voted. This bright-line segregation of information helps protect Privacy I.

  To preserve these protections, audit logs should ensure that information about the votes cast is stored separately from information that may be linkable to voter identity, and should ensure that these two types of entries cannot be linked.

- **Protecting Privacy II.** To protect Privacy II, any log entries that might enable a voter to prove how he/she voted should be segregated or identified so that they can be redacted before audit logs are disclosed to members of the public. In my opinion, it is acceptable to store this kind of information in audit logs, as long as there are sufficient controls on disclosure of the information to protect against coercion and vote-buying. As a general principle, this kind of information should be treated similarly to cast vote records or ballot images, because they share the same properties and risks.

  Cast vote records serve as an existing precedent for how to handle sensitive information while protecting Privacy II. Most jurisdictions do not routinely disclose cast vote records to the public. The reason is that routine public disclosure of cast vote records can enable voters to sell their vote or prove how they voted, through an attack known as "pattern voting." In a "pattern voting" attack, the vote-buyer provides each vote-seller with an individualized voting slate, instructing the vote-seller exactly how to vote in every contest. The vote-buyer selects a special pattern of votes in down-ballot races (e.g., a combination of Yes and No votes in a set of judicial retention races) that is unique to this particular vote-seller, combines this with the vote-buyer's preferred outcome in contests that the vote-buyer cares most about, and obtains a voting slate that is unique for that vote-seller. If the vote-buyer then has a way of obtaining the list of all cast vote records that were cast, the vote-buyer can check to see whether the vote-seller's slate appears on that list. If it does, the vote-seller will be compensated; if not, the vote-seller will be punished. The "pattern voting" attack could also be used by a coercer to coerce a voter into voting a particular way, enabling the coercer to credibly threaten to punish the voter if the voter does not cooperate, if the coercer can obtain the cast vote records. For these reasons, most jurisdictions exercise control over disclosure of cast vote records. I would recommend that, if audit logs contain any information that could enable a voter to prove how he/she voted or identify his/her ballot, then the same controls should be applied to those portions of the audit logs.

  Ideally, a voting system would be designed to record any information that could be used by a voter to prove how he/she voted in a separate file, so that it can be redacted or protected from disclosure and so that the rest of the audit log can be easily disclosed without endangering Privacy II. If the voting system is not designed to store this kind of information separately, and if audit logs contain any information that could be used by voters to prove how they voted, then the voting system vendor should supply a redaction tool that can be used to remove this information (and only this information) from the audit log before it is disclosed to the public.

Most modern voting systems protect privacy by never recording any information that could endanger Privacy I or Privacy II in audit logs. In other words, they ensure that audit logs contain no information at all about who voters have voted for. However, it is possible that future voting systems might record additional information, in which case the requirements above would need to be considered.

There is a potential exception to the principles outlined above for provisional voting on DREs. Any DRE that supports provisional voting must store a code that can be linked to the voter's identity together with the provisional voter's vote, so that the vote can be included in the final count if the provisional voter is determined to be eligible to vote and excluded otherwise. Consequently, in the case of provisional votes cast on a DRE, it is not possible to maintain a strict separation between votes and voter identity. This exception only affects provisional voters who vote on DREs. It is not clear that it affects what may be safely stored in audit logs, even for DREs that do support provisional voting; the code that links a provisional voter's vote to their identity probably does not need to be logged in the audit log, in which case no exception need be made for audit logs.

3.1.4 Are there events that should not be logged because they are irrelevant for diagnostic and forensic audit purposes?

Answer. I do not know. I have not been able to identify any specific examples that seem relevant.

3.1.5 Are there events that should not be logged for any other reason?

Answer. Yes. Voting systems should probably refrain from logging personally identifiable information about voters or election workers that could pose an identity theft or privacy risk for them.

For instance, an e-pollbook device might have access to voters' names, addresses, driver's license numbers, SSNs, date of birth, and other sensitive information about voters. To avoid creating identity theft risks for voters, the e-pollbook should not log sensitive information, like driver's license number, SSN, or date of birth, in its audit log. It is my understanding that no e-pollbook is currently approved for use in the State of California, but I list this example for completeness in case it ever becomes relevant in the future. As another example, if county election workers log in to the central election management system by pressing their thumb against a thumbprint reader, the system should not log a copy of the scanned thumbprint, to avoid creating identity theft risks for election workers. I am not aware of any California county that uses such a method for user authentication, so this example may be purely hypothetical.

3.1.6 Where should a voting system save its logs?

Answer. A voting system typically is comprised of many devices. Each device might generate its own audit logs, which typically would be stored locally on that device during the operation of that device; but after the election, all of this audit log data might be collected and copied to a central repository. Therefore, there are two questions: Where should voting equipment store audit log entries when those entries are initially generated? And, how should audit log data be collected and stored for post-election analysis and longer-term archival?

Voting equipment should record audit log entries onto durable non-volatile storage. Suitable storage media might include flash memory or EEPROM/EPROM. Audit logs could be stored on a removable device (such as a memory card, memory cartridge, CF card, or similar item) or on a non-removable internal component of the device. To my knowledge, every major voting system stores audit log data in this way.

After the election, the voting system should be designed to routinely collect all audit logs from every device used in the election and transfer a copy to the central election management system. At that point, the

collected audit logs should be stored on durable non-volatile storage associated with the election management system. There are several advantages to designing a voting system to routinely collect audit logs from the field in this way. First, once the audit logs are all in one place, it becomes easier to perform post-election analysis of the audit logs, across all polling places, to check for problems that may show up in the logs. Second, having all the audit logs in one place makes it easier to back up the audit logs and archive them for retention as part of the 22-month retention period for election-related materials.

3.1.7 Where, how, with what frequency and by whom should logs be backed up?

Answer. The exact frequency with which audit logs are backed up is an operational question and it seems reasonable to leave this up to the system administrators in charge of the election system. It seems that it would be reasonable to require that audit logs be backed up at least once after every election. It would make sense to archive all audit logs after every election and retain them for at least 22 months, like all other election materials.

Ideally, archived audit logs would be stored on durable, non-volatile, write-once media. CD-R, DVD-R, or DVD+R discs would be an excellent choice for archiving audit logs, because they cannot be accidentally or intentionally re-written after data is burned to the disc, and because they are likely to preserve the data reliably for at least 22 months.

It is important to archive all audit logs after every election, because otherwise they may be overwritten in the next election. It may be reasonable to perform these archival backups after the end of the 30-day canvass and certification process, to avoid burdening election workers with an additional task during the critical 30-day canvass period.

Some individual counties might decide to perform backups more frequently, to reduce the risk of computer failures leading to loss of audit log data. Those extra backups could be stored on any storage media appropriate for data backups; I see no reason why they should be required to be stored on write-once media (though of course system administrators might well choose to use write-once media if that makes more operational sense, given their circumstances). I do not know of any compelling reason to require counties to back up audit logs more frequently than once every election: it seems reasonable to allow counties to decide on their own whether to perform backups more frequently than once every election.

Backups should be performed by county staff. Vendors should provide sufficient documentation to enable county workers to perform these backups, including documentation of the steps needed (if any) to collect all audit logs onto the election management system and the directories that need to be backed up to retain an archive of all audit logs.

In exceptional circumstances, it might be helpful to take additional backups, to preserve evidence. If it is known that the election may be disputed or may be subject to significant public controversy, as a result of allegations of misconduct by county staff, it may be advisable for the head county election official to immediately make a secure backup of all audit logs. In those circumstances it might be prudent to back up the audit logs to write-once media, to sign and date the backup, to store it securely, and to preserve the chain-of-custody of this backup. The purpose of doing so is to preserve evidence, so that in the event of an investigation, allegations can be investigated and resolved. This step may also serve to protect election officials from unfounded allegations of misconduct. Similarly, if it is expected that the election may be disputed or contested in court or a post-election investigation is likely, due to serious voting equipment failure or improper operation of the voting equipment of a serious nature, then in some cases it might be helpful to preemptively back up the audit logs (and indeed, the entire election management system) and preserve this backup for investigators. Since this is not likely to be required except in rare cases, formal procedures to handle these exceptional cases seem unnecessary; whether to perform additional backups can presumably be left to the discretion of county election officials.

3.1.8 What security features are required to protect logs against alteration or destruction?

Answer. It is not clear that strong protections against malicious tampering with audit logs are essential. Since audit logs are not the primary line of defense against deliberate tampering or other malicious activity, protecting them against tampering may not need to be a top priority. Of course, in an ideal world, where cost was not a concern, we might prefer voting systems that provide strong security protections for audit logs. However, since these protections would come at a cost, and since in practice resources are limited, there may be other, better uses of the limited resources available to election administrators and voting system vendors.

The level of protection required may depend upon for what purpose the audit logs will be used. Many uses of audit log data do not require any special security features from the audit logs. It seems likely that mishaps, failures, and unintentional error are far more common than malicious tampering, and it seems likely that there will be many more cases where audit logs are used to diagnose the causes and effects of these mishaps, failures, and unintentional errors than cases where audit logs are used to investigate malicious conduct. Moreover, when audit logs are used to investigate failures or inadvertent mistakes, there is no reason to expect malicious tampering and no need for specific protections against malicious tampering with the logs. Therefore, most uses of audit logs might not require any special security features to protect these logs.

Ideally, audit logs would be stored (throughout their lifetime) on storage media that is not amenable to easy tampering by unauthorized individuals. However, it must be recognized that there will be limits to the level of security against tampering that can be provided.

If a bit more security is desirable, one possibility would be to ask that the logs be stored securely enough that no one person, acting alone but in the presence of others, can tamper with the logs without the tampering being noticed. This might be enough to ensure that no single person can tamper with audit logs, as long as the device is never left unattended or in the sole custody of a single individual.

If stronger security is desirable, it is possible to use advanced cryptographic techniques, such as digital signatures and hash chaining, to protect audit logs from being modified while in transit back to election headquarters and to detect tampering after the logs have been frozen. Existing systems do not appear to use these methods. These cryptographic methods do introduce complexity into the voting system, which comes with its own costs and potential disadvantages.

If it is deemed important to provide strong protection against tampering for audit logs, a sensible approach might combine multiple protections. Audit logs could be stored on media that is subject to physical security measures, to prevent casual tampering (e.g., by voters or poll workers). Voting devices could cryptographic sign audit logs using a public-key digital signature, and could apply hash chaining to ensure that log entries cannot be undetectably once they have been added to the log [5, 39, 32]. The election management system could check the validity of the cryptographic signatures and the hash chain data structure. Audit logs could be replicated on multiple machines, and the consistency of these copies could be automatically checked using methods described in the research literature [38].

Overall, it seems more important to protect audit logs from accidental destruction or alteration than to protect them from deliberate, malicious tampering. See Section 3.2.2 for further analysis about how voting systems might protect against accidental destruction of audit logs.

No voting system device should ever support modification or alteration of log entries once they have been appended to the audit log. This protects against accidental alteration of log entries. It may also provide a limited degree of protection against intentional alteration through unsophisticated means, but it may or may not fully prevent intentional alteration, because there may be ways to bypass the voting system software and alter the audit log directly (e.g., using other installed software not provided by the voting system vendor).

No device should ever support deletion of specific log entries individually selected by an operator. The only reason to delete audit logs is to free up space by deleting logs that are no longer needed, which can be achieved by deleting an entire log wholesale, without providing the functionality to delete single log entries.

The functionality to delete individual log entries destroys the integrity of the audit log as a whole and can be too easily abused.

3.1.9 What audit log reports should a voting system be capable of producing?

Answer. As a rule of thumb, there is little point to an audit log that will never be examined; audit logs that no one ever looks at are of little benefit. Therefore, a voting system should ideally be able to produce reports that assist audit log users in making good use of audit logs.

To facilitate post-election investigations, a voting system should at a minimum be able to produce the following reports:

- **Audit log export.** It should be possible to export all of the audit logs to an open format, suitable for use and analysis by others. See Section 3.2.3 for further discussion of how this can be best achieved.

- **Vote tallies report.** It should be possible to produce a report, listing the vote tallies for each candidate, broken down by precinct. It should be possible to further break down these tallies in each precinct into several categories: e.g., ballots cast at the polling place on election day; absentee ballots cast separately; provisional ballots cast at the polling place and accepted later. This report should be available in an open format, suitable for use and analysis by others. See Section 3.2.3.

Preferably, a voting system should also be able to produce the following reports:

- **Anomaly report.** It should be possible to produce a report summarizing any anomalous events that appear in the audit logs. This might include machines that were opened substantially later than the official time when polls should open; machines that were closed earlier than the official time when the polls close; cases where poll workers printed results reports early (before the closing time); or any sequence of events that should be impossible or that indicate a failure of procedure or equipment.

- **Machine allocation report.** It should be possible to produce a report indicating, for each voting machine, which precinct or polling place it was assigned to. This report should be available in an open format, suitable for use and analysis by others. Such a report would be helpful, because it provides a way to trace back any problems identified in the audit logs to the precinct or polling place that was affected.

- **Cast vote records.** It would be useful if the voting system could also export all of the cast vote records (also known as ballot images) to an open format, suitable for use and analysis by others. Each cast vote record should be traceable back to the machine on which it was cast. Of course, not all voting systems save cast vote records; for instance, many older precinct-count optical scan machines save only vote totals and not the individual cast vote records. Obviously, if the system does not record cast vote records, then it will not be possible to export them. However, export functionality should be provided for all cast vote records that are saved.

These reports would assist post-election investigations and forensic and diagnostic audits.

If we want election officials to be able to use audit logs for routine post-election assessment after every election, then election officials will need some way to obtain reports that analyze and summarize audit results. It is not clear to me what kind of summary information would be most useful to election officials. The best way to determine what kind of reports would assist election officials would be to involve election officials, who know what they would find useful, in a discussion with technical experts who are aware of what could feasibly be provided. While I am not qualified to determine what election officials might find useful, here are some possible audit log reports that a voting system could potentially produce:

- **Polls opening/closing times.** It would be possible to produce a summary report that provides statistics on the time at which the polls were opened and closed in each polling place, according to the audit logs produced by polling-place devices. Such a summary report might identify the number of machines or polling places where the polls were opened at least 30 minutes after the official election start time, and where the polls were closed before the official election closing time, so that officials can gain a sense of how widespread problems with opening/closing the polls may have been. The report might optionally list those precincts, so that officials could look into those precincts and use this information to identify sites where poll workers struggled to meet their duties.

  The summary report could also provide a histogram or chart of the poll-closing time, according to audit logs from devices in each precinct. This information might provide a way to get a feeling for how many polling sites had to stay open late to meet voter demand, and how long lines were at the official time for closing the doors to the polling site. Thus, such a report might be helpful for making decisions about where to allocate machines in future elections.

- **Machine failures.** Analysis of the audit logs can identify some kinds of problems with voting machines, such as voting machines that are turned off before the close of polls (e.g., because they were not working properly) or voting machines that experienced a low-battery condition. A summary report could list how many precincts experienced problems with their machines, and identify the precincts that experienced the most severe problems (e.g., that had the greatest loss of machines, as a fraction of their total allocation, over the greatest period of time). While this would not identify all machine failures, it might identify some kinds of failures.

- **Detection of anomalies.** It might be possible to build an audit log analysis tool to identify anomalies in the audit logs: e.g., machines whose audit log entries appear anomalous and out-of-the-ordinary for what would be expected during normal operation. It might be possible to produce a summary report identifying those precincts and machines, so that election officials can investigate further, e.g., by checking for trouble reports from poll workers and looking through the materials collected from that polling place for any evidence of problems.

- **Mismarked ballots.** For a precinct-count optical-scan system, if the audit logs record an event each time the scanner detects a mismarked (e.g., overvoted) ballot and returns it to the voter, it would be possible to produce a summary report identifying the rate at which this occurs. If the audit logs also record an event if the voter or poll worker chooses to override the warning from the scanner and submit their ballot anyway, it would be possible to produce a summary report identifying the rate at which the scanner's warnings are overridden and kicked-back ballots are re-submitted. This might provide some insight into poll worker behavior. Many jurisdictions train poll workers to encourage voters to re-mark their ballot if the ballot is rejected by the scanner. The summary reports mentioned above might provide some indication about the extent to which this training has been successful.

- **Compliance with certain procedures.** It may be possible for analysis of audit logs to produce a crude measure of the extent to which poll workers have followed certain procedures. Poll workers are required to follow many procedures; in some cases, violations of these procedures leave a telltale sign in the audit logs. For instance, as mentioned above, if poll workers are unable to set up a voting machine on time, or if a machine is shut down early, this will be apparent in the audit logs. As another example, as mentioned above, if poll workers are not consistently encouraging voters to re-mark ballots kicked back by a precinct scanner, but rather are simply overriding all machine warnings as a matter of course, this may be detectable from the audit logs. Also, at the end of the day, poll workers are usually required to print at least two copies of the summary report (one for display outside the polling place, one for return to county headquarters); if poll workers fail to print the required number

of summary reports, this may be discernable from the summary report. Alternately, if poll workers improperly print summary reports early, before the close of elections, this may be detectable from the audit logs. As a fourth example, in California several voting systems are subject to a special use condition: counties may provide an accessible DRE machine in each precinct for voters who are unable to vote on paper ballots, subject to the requirement that if at least one person votes on the accessible DRE, then poll workers must ensure that a a minimum of five people vote on the DRE. It would be possible to discern from audit logs whether this use condition has been met.

Based upon these indicators, it would be possible to construct a metric that aggregates these indicators of procedural failure to obtain a numerical score for each polling place. A summary report could then summarize these scores by precinct, perhaps displaying a color-coded map where colors are used to indicate the score for each precinct. A summary report could also indicate which procedural requirements are most consistently followed and which ones are most frequently violated. Election officials could conceivably use this metric to assess poll worker effectiveness, perhaps paying a bonus to poll workers at precincts where the score was above-average, or perhaps targeting those poll workers as a top priority for recruiting in future elections.

It is not clear whether this concept would be useful or advisable. Audit logs can only provide a narrow and limited view into the effectiveness of poll workers. Also, audit logs provide visibility into compliance with only a small fraction of the procedures that poll workers must follow, so this metric would necessarily be crude and limited in applicability.

- **Misuse by election staff.** A summary report could flag signs of misuse by operators of the election management system, if any are present in the audit logs. For instance, some jurisdictions have regulations prohibiting election workers from printing or accessing vote tally reports before the close of polls on election day. If the election management system's audit log records all accesses to vote tally reports, then it would be possible for an audit log analysis tool to check for any sign that workers have violated this regulation, and if so, highlight that fact in a summary report produced by the county's chief election officer.

  As another example, many jurisdictions have a policy that each election worker must sign into the election management system using their own computer account, to provide accountability. If the audit logs indicate that a single person was simultaneously signed in from more than one machine, and both sessions were concurrently active, this could potentially indicate a violation of that policy. (It could also be benign.) A summary report could identify any detected instances of simultaneous logins.

  It might be useful if the voting system could provide an activity report summarizing the activities of each of the authorized operators of the election management system. Such a report might summarize the categories of functionality accessed by each individual (e.g., election definition, ballot layout and design, report generation) as well as the type of access (e.g., read-only access vs. write access) to that functionality. The report might also summarize the dates and hours of activity for each worker, providing a way for the chief election officer to see at a glance which workers have accessed the system and perhaps to detect any irregular accesses.

- **Machine allocation.** It might be possible to infer some information about the degree of utilization of the machines in each polling place, identifying polling places where the machines are especially heavily utilized or especially lightly utilized. For jurisdictions that use DREs as their primary means of voting, this information might be helpful in adjusting the allocation of machines to polling places in future elections.

  Typically, audit logs from voting machines identify the time at which each ballot was cast. From this, it should be possible to estimate the rate at which ballots are cast at each machine, and how this rate

varies throughout the day. For instance, it would be possible to identify the times of peak utilization and the duration for which each peak-utilization period lasted. If most polling places experience a peak from 7am to 8am, but at one polling place the utilization rate for each of the machines in that polling place remains at its peak from 7am to 9:30am, that might be a sign that the polling place might benefit from more machines. Conversely, if the utilization rate at one polling place remains consistently low, that might be a sign that the polling place might have received more machines than needed—or it might be a sign that the voter sign-in process was overwhelmed and more poll workers would have been useful.

This concept is highly speculative and to my knowledge has never been tested. While it is plausible the additional information provided by audit logs might improve the ability to refine the allocation of machines, I am not aware of any test to see whether this is indeed the case or not.

I emphasize that many of these possibilities are speculative and have not been carefully tested. Also, I do not know whether any of these kinds of reports would be useful to county election officials or not. I am not suggesting that a voting system needs to support all of these reports, or that failure to support these reports makes a voting system deficient. Rather, the list above is intended to serve as a sample of some ways that appropriately designed summary reports might be able to assist election officials in their duties.

3.1.10 What features are necessary/desirable to make audit logs usable?

Answer. The answer depends upon the category of user who will be using audit logs:

- **Election officials.** Election officials will probably need tools to analyze and summarize the audit logs for them. Audit logs contain a great deal of raw data, which is unlikely to be directly useful to election officials in its raw form. For this reason, the most important feature to election officials will probably be the availability of summary reports, such as those discussed in Section 3.1.9, which extract actionable information from the raw audit logs.

- **Citizen observers.** Members of the public and election observers will probably have two essential needs. First, they will need access to the audit logs. This means that audit logs must be designed so that they can be safely released to the public, and the voting system documentation must clearly state that this is the case, so that election officials feel comfortable giving observers access to the audit logs. This may be a significant barrier with existing systems.

  Second, audit log data will need to be available in an open format that is suitable for use and analysis by others. See Section 3.2.3 for discussion. Observers might need documentation on the format of this file and the meaning of codes and fields in this file, if they are not self-documenting.

  Observers will likely benefit from access to any part of the voting system documentation or Technical Data Package that discusses audit logs.

  Observers will likely find it very helpful to have access to the kinds of reports mentioned in Section 3.1.9 as useful for post-election investigations.

  Observers might find it useful to have tools to analyze and summarize the audit logs. However, I do not classify this as a necessary feature: technically sophisticated observers may be able to develop some of those tools on their own, even if they are not provided as part of the voting system.

- **Election contests.** In litigation over election outcome, it is likely that candidates will hire election lawyers and expert witnesses to analyze the evidence for them. These individuals will share all of the concerns as citizen observers: they will need full access to the audit logs, in an open format amenable to analysis. They will also likely need access to the reports mentioned in Section 3.1.9 as useful

for post-election investigations. They might find log analysis tools useful, but their expert witnesses might also develop their own custom tools in any case. Litigators may have other concerns that I am unfamiliar with.

- **Criminal investigators and prosecutors.** It is likely that criminal investigators and prosecutors will share many concerns with election litigators: they will need complete access to the logs and all election-related records, in a form that is suitable to analysis, and they may need system documentation relating to the audit logs and other reports.

  Criminal investigators might need access to much more than just the audit logs: e.g., all electronic records on the election management system, operating system logs (not merely those generated by the voting system itself), video camera surveillance records, and possibly more. It is conceivable that criminal investigators might prefer to seize the entire election management system, or make their own bit-for-bit copy of its hard disk, rather than relying upon election officials to produce reports for them. Criminal investigators may also have concerns regarding the chain of custody and authentication of the records. I do not have enough experience to predict how this might affect what features criminal investigators and prosecutors need the voting system to provide.

  Because criminal investigators are trusted government employees, concerns about confidentiality of the information and whether it is safe to disclose to the public may be less relevant. I expect criminal investigators will demand access to unredacted audit logs.

3.1.11 What features are necessary/desirable to make audit logs accessible to persons with disabilities?

Answer. For the kind of audit logs considered in this report, I would expect that no special steps will be necessary.

Voter-verified paper audit trails (VVPATs) also have the potential to raise accessibility concerns for voters with disabilities, but VVPATs do not fall within the category of audit logs considered here and thus are outside the scope of this study.

3.2 Additional considerations

3.2.1 Audit logs should be a real-time, immutable, append-only log

The voting audit log should be stored as an append-only structure. As each loggable event occurs, an entry should be immediately and irrevocably appended to the append-only log on non-volatile memory. Each append operation should be irreversible. Log entries should be immutable: there should be no way to modify them after they have been appended to the log.

Audit log entries should be stored explicitly as soon as they occur. The contents of the audit log should not be inferred or computed post-facto from other data structures. Reconstructing or re-creating on demand a plausible sequence of events that could have occurred, inferred based upon the contents of other data structures (such as vote tallies), does not qualify as an append-only audit log; such an approach would eliminate many of the benefits of a true audit log.

3.2.2 Audit logs should be protected from accidental destruction

To help prevent accidental destruction or deletion of audit logs, the voting system should avoid creating features that could potentially cause audit logs to be inadvertently deleted.

One of the best ways to protect logs from accidental destruction is to follow a general principle of striving to avoid ever deleting audit log data. Audit logs should be append-only; voting system software should be written to prevent modifying or deleting individual log entries, and should be written to avoid deleting log data except where necessary. The one case where it may be necessary to delete audit log data is where the storage medium does not contain enough space to store audit log data from past elections. This seems unlikely to apply to election management systems.

Given modern technology, as far as I can tell there appears to be little or no reason to ever delete audit logs stored on a central election management system. Based on audit log data I have been able to obtain, I estimate that it takes on the order of magnitude of 1 Megabyte of storage per election to store all of the audit logs from all polling-place devices, if stored in compressed form. Today, hard drives cost about $0.10 per Gigabyte, or about one-hundredth of a cent per Megabyte—which corresponds to a cost of about one-hundredth of a cent per election to store a copy of the audit log data on the election management system. Consequently, the cost of storing a copy of all audit logs, from every election ever conducted using that system, on the election management system appears likely to be negligible. Based on this, I conclude that there may be no need to ever delete audit log data from election management system machines.

Therefore, it appears that election management software could plausibly be designed to avoid deleting or modifying of audit log data under any circumstances. In such a design, software developers would deliberately avoid introducing any feature for deleting the audit logs; that simply would not be a supported operation. Such a design would help avoid prevent accidental destruction of audit logs.

In contrast, there is a good justification for components with limited storage capacity to support deletion of audit log data. In currently deployed voting systems, many polling-place devices—including precinct-count optical scan machines, touchscreen machines, and voter card activators—have only a limited amount of non-volatile storage, and the amount of storage in these devices cannot be easily upgraded. However, it might be possible to provide several protections against unintentional deletion of audit logs on these devices. Some devices could be programmed to always save audit logs from the last election in addition to audit logs from the current election. In some cases, it may be possible to design the devices to only allow deletion of the audit logs after the audit logs have been uploaded onto the central election management system or exported for transport to the election management system. Another possible approach might be to always retain audit logs on the device for at least 22 months, or until the device uploads its logs to the central election management and receives an acknowledgement from the election management system that those logs have been received and archived. These features might help prevent accidental destruction of audit logs.

Voting system evaluators may want to check whether polling-place devices expose any functionality that would allow poll workers or rovers to delete audit logs. If they do, it may be prudent to examine carefully whether this functionality is necessary and appropriate. Giving poll workers and rovers the ability to delete audit logs increases the risk of accidental or improper deletion of audit logs. Given the experience in Webb County, Texas in March 2006 of voting machines whose votes and audit logs were deleted in the middle of election day, it seems reasonable to think carefully about what functionality is exposed on election day to poll workers and rovers. It may well be safer to design voting systems so that audit logs can only be deleted by county election workers, and not by poll workers or rovers.

3.2.3 Audit logs should support open file formats

To enable third-party analysis of log files, it is important that audit logs be available in an open file format. By open file format, I mean a file format that can be read and parsed without proprietary tools and without access to any proprietary or non-public information. Normally, this would mean that the file format is openly documented, or is so simple as to be self-documenting. If the voting system stores audit logs internally in a proprietary file format, it must provide some way to export all of that data into an open file format, without

loss of information.

Also, it is important that the data be available in a form that is amenable to automated analysis. It must be possible to build tools to read the audit log entries and programmatically analyze them.

For instance, suitable file formats might include a plain text file, where there are no undocumented codes or fields (for instance, each line of the text file might represent one log entry), an XML file, using a publicly disclosed schema (DTD) where the meaning of every tag and attribute is publicly documented, or a spreadsheet saved in CSV (comma-separated value) format, where the meaning of each column of the spreadsheet is fully explained in a public document. A PDF document is unsuitable, because it is not amenable to automated analysis: there is no easy way to extract data from a formatted PDF document. Similarly, a paper print-out, or a scan of a paper print-out, is unsuitable, because it is not amenable to automatic analysis.

Open file formats are beneficial for several reasons. First, if audit logs are to be disclosed to election observers, members of the public, election investigators, criminal investigators, forensic auditors, or others, then they will need to know how to interpret the data that is provided to them. The best way to do that is through an open file format. In this way, open file formats can promote transparency and make audit logs more useful to consumers of audit log data. Second, open file formats support interoperability: they enable third parties to provide audit log analysis tools that offer functionality not present in the software provided by the voting system vendor.

3.2.4 Audit logs should be publicly disclosable

Audit logs should be designed so they can be safely disclosed to election observers and members of the public. They should avoid including information that cannot be publicly disclosed, whether for reasons of security or privacy. Publicly disclosable audit logs support transparency by enabling election observers to use the audit logs to gain insight into the effectiveness of the processes, procedures, and technology used in the election.

The format of audit log files must be documented in a specification that is freely available to the public. The meaning of all codes and log entries must be clearly explained. The audit logs, and accompanying documentation, must be designed to enable candidates, political parties, election observers, the media, and members of the public to understand and make sense of the audit logs. These specifications must also be detailed enough that individuals can develop their own software tools to parse and analyze the audit logs.

Professor Doug Jones recommends that all audit information should be made public, "with the exception of that information that can be shown to be a danger to the integrity of the election process". He further recommends that "security analysis of the voting system should clearly identify these specific pieces of information in advance of the use of that system" and that "the burden of proof should be placed on the election authority to demonstrate that nondisclosure is required, and the amount of material withheld and the duration of withholding should both be minimized" [28].

Chapter 4

Evaluation of existing systems

This chapter analyzes each voting system considered this study, to assess how well it meets the criteria listed in Chapter 3.

Caveats. This analysis is based solely upon review of the documents available to me, including questions submitted to the voting systems vendors and their responses and other public documents. I did not have access to a working voting system or audit logs produced by the voting system to test my hypotheses and conclusions. While I used my best professional judgement and examined these documents diligently, in many cases the nature of this study makes it difficult to draw definitive conclusions about these voting systems. Therefore, all conclusions and opinions expressed here should be viewed as tentative and unconfirmed.

This analysis is intended to assist in evaluating the extent to which existing voting system audit logs support the goals of public elections, and to identify ways in which future voting systems might potentially be able to improve this support. The purpose of this analysis was not to assign blame or liability or to pass judgement on the competence of voting system designers. Voting systems that are deployed and used today were designed many years ago. Since technology has advanced significantly since these voting systems were designed, and since our understanding of and experience with voting system audit logs has increased as well, some design considerations have changed over time, and the design choices that made sense at the time these systems were designed may no longer be the best choices for the future. As a result, it seems appropriate to evaluate today's voting system audit logs with an eye towards how well they meet their goals and how they might be improved.

4.1 Common features of all six systems

There were a number of features that appear to apply to every voting system I examined, or almost every voting system. In particular:

- None of the systems provided publicly available documentation on the format of audit log files.
- None of the systems provided tools to analyze the audit log files and produce useful summary reports highlighting the most important information in the logs.
- Based on the documents available to me, I did not see clear signs that the vendors had exhibited technical leadership to think through how voting system audit logs can be most useful and deliver this in their product. Instead, it appeared that vendors may have treated audit logs primarily as a compliance exercise, focusing on meeting the minimum requirements in the federal voting standards.

This is not surprising, given commercial pressures and the structure of the voting system market; it is not surprising that commercial companies would focus on delivering what their customers are asking for. Nonetheless, my review of the documents available to me suggests that we should not look to the vendors alone as the source for vision and leadership on audit log functionality.

- In every system I examined, I expect that an operator who has authorized access to the election management system and is knowledgeable, motivated, and malicious may be able to bypass the security controls included in the vendor's election management software and directly access, modify, or delete audit log entries. Because election management systems are general-purpose PCs with the capacity to execute arbitrary code, and because COTS operating systems are generally not designed to protect against a malicious operator, I expect that an operator with access to the system can likely do anything that the election management software can do, including modify or delete audit logs. I am not aware of any simple or easy way to eliminate this shortcoming; this property appears to be a consequence of the way that these systems are designed and architected. If the voting system is designed and configured with sufficient care, it may be possible to prevent casual tampering and attacks by unsophisticated users, but on general principles I expect that it will be challenging to completely prevent attacks by knowledgeable, sophisticated users given the architecture of today's voting systems. These expectations are supported by the security analyses conducted as part of the California Top-To-Bottom Review. In particular, these reports concluded that trusted insiders would be able to bypass access control measures and tamper with audit logs and other system data for the ES&S [20, §3.9–3.11] [4, pp.8–9], Hart [2, §III.1] [26, §6.5, 6.6], Premier [1, §III.1.a–c] [9, §5.3], and Sequoia [8, §1.4, 3.3.1, 4.2.10] [21, §5.1] systems.

  In a number of cases the vendor documentation appeared to be overly optimistic about the level of protection provided against these threats. In some places the documentation suggested that it would be impossible for operators to tamper with the logs. In my view, such assertions are dubious and should not be accepted without rigorous evidence. In some cases, vendor documents appeared to assume that if the log file format is not text-based, then tampering with or modifying audit log entries is not possible; however, this assumption is not justified.

  It could be debated whether this shortcoming of the voting systems is serious or not. There are two reasons why this shortcoming may not be as serious as it sounds. First, most uses of audit logs are likely to concern technical failures or inadvertent mistakes. Second, today's voting system software is generally unable to prevent trusted insiders from taking many malicious actions anyway, including subverting the election management software or replacing it with malicious code; the consequences for audit logs seem relatively minor when compared to the other potential consequences of this property of today's voting systems. In those exceptional cases where malicious behavior by trusted operators of the election management system is suspected, voting system audit logs should not be treated as beyond suspicion.

4.2 DFM BCWin

Summary. Only limited information about the BCWin system was available to me, so I was not able to draw conclusions about the extent to which BCWin's audit logs meet the criteria for audit logs outlined earlier, or about the strengths and weaknesses of BCWin's audit logs.

Other notes. The BCWin system has never been submitted for review under the federal voting standards and I do not know whether it complies with the requirements governing audit logs found in the federal standards.

BCWin stores its audit logs in a separate directory from the election database. The vendor mentions 6 audit logs: Card Reader Logs, Transmission Logs, Application Log, Utilities Log, Main Server Application Log, and Client Workstation Counting log. It is not clear to me exactly where these are stored. I was not able to determine whether the BCWin documentation describes how to back up audit logs, and whether all 6 logs are archived as part of the ordinary, documented method for archiving the election database.

It is not clear whether documentation about the design, characteristics, and format of the BCWin audit logs exists. In response to a request for "any manuals or other written materials available to technicians and programmers concerning the characteristics and use of audit logs", the vendor responded "N/A."

4.3 ES&S Unity

Summary of strengths. All major system components generate audit logs. The ES&S system incorporates a separate component specifically dedicated to managing audit log files produced by certain election management software components.

The ES&S system appears to have good support for managing audit logs from iVotronic DREs and exporting the audit logs and ballot images (cast vote records) from iVotronics into a text format that is amenable to third-party analysis. However, the iVotronics are not presently used in the State of California.

Summary of weaknesses and concerns. The ES&S system does not appear to provide good support for collecting, managing, analyzing, archiving, and exporting audit logs from the M100 precinct scanner, M650 central scanner, and AutoMARK ballot marking device. As far as I could tell, it appears that the system does not automatically upload audit logs from these components to the central election management system, as part of routine operation. The system does not provide functionality to analyze or summarize the audit logs.

Many of the ES&S audit log files are recorded in a proprietary format that is not publicly documented, and in several cases, the file formats do not appear to be documented even in the confidential TDPs. I was not able to find any documented way for a county to export audit logs from the M100 precinct scanner, M650 central scanner, or AutoMARK ballot marking device into a text format that is is amenable to third-party analysis.

Detailed analysis. The ES&S system includes a separate component, the Audit Manager, specifically for managing audit logs. This component can be used to view audit logs, print them, or export them to a CSV-formatted spreadsheet [40, §2.5.6]. The ability to export to CSV format is a strength of the system.

I was not able to determine from the documentation whether logs from all polling-place devices are normally uploaded to and available from within the Audit Manager; there was no indication that they are. It appears that the Audit Manager can only be used to access audit logs generated by the Election Data Manager (EDM) and ES&S Image Manager (ESSIM) components, and in particular, the actions taken by users of the EDM and ESSIM components. I was not able to find any sign that the EDM audit logs include logs generated by the M100 scanner, M650 scanner, or AutoMARK ballot marking device.

The ES&S Audit Manager specification is marked "Company Confidential." I was not able to find publicly available documentation on this system component. In my view, this is a weakness in the system documentation.

Based upon review of the ES&S TDPs, it appears that the system creates many different log files, each with its own file format:

- The M650 central scanner creates .LOG files holding the M650 audit logs. The software specification for the M650 scanner in the TDP provides a lengthy list of the possible types of log entries, with a brief explanation for each; this is a strength of the documentation.

The M650 .LOG files are apparently saved in an "encoded format", but the format is not specified in the TDPs. ES&S explained in response to questions from the California Secretary of State's office that

> *ES&S:* [The .LOG file] is not a text file but an encoded file that requires an algorithm to decode it. The only available ES&S software that can print this log is the M650 itself.

In my view, this is a weakness in the M650.

Based upon M650 system documentation, it appears that the M650 can print the decoded audit logs but cannot export them to a file for third-party analysis or upload them to the central election management component. In my view, this is a weakness in the M650.

- The M100 precinct scanner stores audit logs in a binary format on the PCMCIA memory card inserted into the M100 scanner. The file format is documented in the software specifications for the M100 scanner [41, pp.151–155], but the file format would not be self-evident without access to the specifications. The M100 software specifications are not marked "Company Confidential".

  Log entries are stored as code numbers. I was not able to find documentation of all possible codes and their interpretation. Without documentation of the file format and the meaning of codes, it would very difficult to make sense of these log files. The M100 software specifications reference the documents "Election Generating Software Error Messages" and "Precinct Ballot Counter Error Messages" as specifying the possible error messages, but I was not able to find those documents. ES&S writes:

  > *ES&S:* For the M100 there is no file structure used on the M100's PCMCIA card. There is a binary image on the card that contains the election definition, the results, and the log information. All are embedded in the data structure of the image. The firmware and the applicable software that reads the card analyzes the header information looking at field sizes to determine data locations.

  I was not able to determine whether the ES&S system comes with any tools to decode the M100 audit logs and export them into an electronic format amenable to automated analysis.

- The Data Acquisition Manager (DAM) creates .SUV, .SPR, .SPH, .SPM, and .SVI files holding various kinds of log data. These files are apparently stored in a text-based file format, though the documentation available to me does not appear to provide examples of these files.

- The Election Reporting Manager (ERM) apparently stores audit logs in .ALG files. According to ES&S:

  > *ES&S:* The log file is not in textual form and is created by the COBOL runtime which encodes the stored information. It requires the COBOL to interpret the stored data.

  It is not clear whether there is any way to export the entries in this file into a format that can be interpreted by others. The TDP contains a section that purports to document the .ALG file format, but I was not able to understand the contents of that section. I infer that it probably would not be possible for customers to make sense of this file on their own, without ES&S's assistance. In my view, this is a weakness in the system.

  Apparently, there is also a System Log and a Manual Entry Log. It is not clear what the System Log stores. The Manual Entry Log is described as listing "all the totals entered or changed manually." These two files appear to be stored in a text format, but no sample log files were provided in the

TDPs. I was not able to ascertain whether these are audit log files or whether they are reports that can be produced from the log files themselves.

It appears that the files ERSLOAD.TXT, DATALOAD.LOG, ERSAUDIT.LOG, and LDETAIL.TXT may also contain log information. I was not able to determine what these files pertain to.

It appears that the Election Reporting Manager has the ability to produce several reports that may contain audit log data, under the filenames EL68A.LST and EL68.LST. I was not able to find any description of the contents of these reports or samples of these reports in the TDPs.

- The Audit Manager stores log files in the EALSYS.MDE database.
- The Hardware Programming Manager stores log files in .ELG files. This file is not in a text-based format.
- The AutoMARK ballot marking device stores audit logs in files named OP.ELG and SCAN.ELF. ES&S explains:

  > *ES&S:* Both files are encoded and cannot be viewed in text format. The log can be printed by the [AutoMARK]. There is an internal ES&S utility that has not yet been made available that can convert the log to text form.

- The AutoMARK Information Management Software audit logs are stored in a separate SQL database.
- The iVotronic terminal (not presently used in California) stores audit logs in .EVT files.

In short, the ES&S system produces many audit log files, and there is no good integrated way to collect or manage all of these logs.

In addition, the ES&S TDPs appear to provide only minimal information about the format, nature, and interpretation of the audit logs. Much of the information that is available is marked "Company Confidential", and there does not appear to be any publicly available specification of the content or format of ES&S audit logs.

The ES&S response to a question from the California Secretary of State's office about the possibility of deleting/modifying audit log entries raises questions about whether authorized users are able to delete or modify audit log entries:

> *CA SOS:* Is it possible to delete or modify any audit log entry or entries?
>
> *ES&S:* [...] The tabulators and ballot marking devices provide audit logs through external media and printouts. This requires all authorized users to follow the correct certified election procedures to prevent deleting or modifying of any audit log entry or entries.

On the other hand, if an authorized user fails to follow the procedures, either inadvertently or deliberately, there appears to be a risk that operator actions might cause audit logs to be deleted.

The source code report for the ES&S system produced as part of the California Top-To-Bottom Review identified several concerns relating to the protection of audit logs against tampering. In particular, the report states that the M100, M650, Audit Manager, Hardware Programming Manager, and the Election Reporting Manager component do not provide effective protection to prevent or detect tampering with their audit logs [4, p.9].

4.4 Hart

Summary of strengths. All major system components generate audit logs. The Hart system provides support for automatically collecting audit logs from all eScan, eSlate, and Judge Booth Controller (JBC) units in the field, as a routine part of election operations. This support is effective if the system is configured and used in either of the following two ways: (a) the Tally component is used to upload MBBs, but the Rally component is not used; or, (b) election officials use the Servo component to "back up" all eScans, eSlates, and JBCs. Therefore, there are two different ways that election administrators can collect all audit logs as part of the routine operation of the system, providing election administrators with a choice.

The Hart system appeared to provide the most complete logging of any of the voting systems I examined. The eSlate, eScan, and JBC log events under many conditions. The system documentation was written in a manner that is detailed and helpful for computer scientists and technical experts.

Summary of weaknesses and concerns. Collection and archival of audit logs has some weaknesses if the Tally and Rally components are used together. It appears that if Tally is used together with Rally, audit logs may not be collected to any single location, and the standard archival processes may fail to archive these audit logs. There are some concerns that Servo's processes for "backing up" and for resetting eScans, eSlates, and JBCs may be error-prone and may lead to inadvertent deletion of audit logs.

I was not able to determine whether the system provides a way to export these audit logs in an open format suitable for automated analysis by third parties. The documents that describe the audit logs are either marked "Confidential" or were not available. The system does not appear to provide functionality for analyzing and summarizing the audit logs and any anomalies found within them.

Detailed analysis. It appears that the Tally component automatically collects audit logs from all memory cards (MBBs) that are read directly into Tally [23, §5.5.3]. This is positive, because it means that so long as Tally is used directly (without Rally), then audit logs will be automatically collected at a central location.

It appears that if Rally is used to collect votes at regional vote centers and upload them to Tally, audit logs from eScans, eSlates, and JBCs may be collected by Rally but not uploaded to Tally. Moreover, it appears that the standard archival processes may fail to archive these audit logs and the system documentation does not clearly describe the steps that must be taken to ensure that all of the audit logs are archived, when the system is used in this configuration [23, §5.5.3.1.1]. As a result, it may be easy for election administrators to inadvertently fail to archive audit logs, if using Tally with Rally.

It appears that Hart's Servo component automatically uploads audit logs from every eScan, eSlate, and Judge Booth Controller (JBC) component as part of the ordinary operation of the system, if Servo is used to "back up" every eScan, eSlate, and JBC after the election. This is a positive aspect of the system.

During the California Top-To-Bottom Review, the Hart document review team reported a concern that it is easy to inadvertently reset devices and delete the data stored on them, including audit logs. Quoting from the document review report [23, §5.5.2.2.2],

> the audit trails from the devices may be lost by accident when resetting devices with SERVO. We observed that when resetting devices with SERVO, the system provides no warning or confirmation for "reset". Moreover, it retains the "reset" checkbox when moving from one device to the other. When one device (such as a JBC) is disconnected and another is connected, SERVO automatically resets the next device. This facilitates the rapid resetting of multiple devices. Unfortunately, however, "reset" and "backup" are part of the same user interface widget. A likely use scenario is the election official user who wants to connect a JBC and just backup the contents of the internal memory, then disconnect it; and then repeat the same procedure on the next device; and so on. This user must make sure that "reset" is unchecked after connecting—or

the device is automatically reset without backing up. Particularly if the user combines backup and reset operations, it would be very easy to accidentally reset without backing up. This could significantly affect the creation of auditable backups, simply through poor user interface design.

In my review of the Hart Servo operator's manuals, I found a clear description of the meaning of the "reset" and "backup" checkboxes. The manual makes it clear that these two options can be selected or de-selected independently. However I did not find a clear and unambiguous warning of the risk of accidental device reset. The operator's manuals do not identify the risks of retaining the "reset" checkbox setting from one device to another, and they do not warn operators to check the "reset" checkbox carefully to avoid unwanted device resets. I was not able to independently confirm the findings of the document review report that audit logs could easily be accidentally lost, but the Servo operator's manuals appear to be consistent with the statements in the document review report.

As the Hart document review report identifies, this risk (assuming it is still present) is potentially mitigated if election officials use Servo to "back up" all devices. However, there remains a risk of inadvertently resetting a device while attempting to back it up, because "reset" and "back up" share the same user interface element. Therefore, this mitigation does not completely eliminate the risk. Also, this mitigation relies upon election administrators to "back up" every device using Servo.

Another possibly mitigating factor is that, according to the Servo functional specification, Servo is supposed to generate an audit log entry whenever Servo is used to reset a device. However it is not clear from the documentation available to me what information is included in this log entry. It is possible that this might provide a way to detect deletion, though it might be difficult to distinguish accidental deletion from intentional deletion, and it might be difficult to retroactively identity the precise impact of an accidental deletion or recover from it. For this reason, this mitigating factor may not be of much comfort should audit logs be accidentally lost due to user interface issues in the Servo component.

In retrospect, it might not make sense to provide a way to reset a device without backing up its contents. This decision decision creates an opportunity for error that may not have been necessary. I am not aware of whether there is any system requirement that would justify supporting reset-without-backup operations or whether there are other considerations that would support this design decision, but it might make sense to re-examine this design decision in future systems.

Servo also provides the capability to reset all data on a memory card (MBB). This is a separate function from the ability to reset an eScan, eSlate, or JBC device. The MBB reset function clears the audit logs and vote data on the MBB. The Servo functional specification states that Servo logs this event and also records the MBB serial number, election identifier, the number of cast vote records cleared, and the number of audit log records cleared. This is a positive aspect of the Servo component, because in case of accidental deletion of a MBB it provides a way to determine how many cast vote records and audit records were lost and which election was affected.

Finally, Servo provides a second way to reset all the data on an eScan, eSlate, or JBC device, through a method dubbed an "administrative reset." This reset does not save or back up audit logs or cast vote records, and so could cause loss of important data if used unwisely.

To archive all audit logs, it is not sufficient to back up the database: several audit log files are stored separately and must be archived individually. In their responses to questions from the California Secretary of State's office, Hart documented the paths to multiple audit log files and database transaction log files that would need to be archived separately. These files include `bossdata.log` (BOSS audit logs), `TallyData.log` (Tally audit logs), and `BN_Default.log` (Ballot Now audit logs). It is not clear whether these three files include the Rally or Servo audit logs as well.

The Hart operator's manuals describe how to archive election data, but the processes documented in the operator's manuals does not appear sufficient to back up all audit log data, including all files listed above. The Hart "Product Description" manual states that copying the Tally database to a CD-R disc is sufficient

to archive all election data. Based upon Hart's responses mentioned earlier, it appears that backing up the Tally database is not sufficient to archive all audit logs: as best as I can tell, this step does not archive BOSS and Ballot Now audit logs, and the status of Rally and Servo audit logs is unclear. The Hart Tally operator's manual has similar shortcomings. The Tally operator's manual describes how to back up the Tally database, but does not explain that this is not sufficient to archive audit log data from other system components. There is no warning of the need to back up other directories as well. The Tally operator's manual does not describe precisely what information this process does and does not save, but based upon a figure it appears that this process backs up the `TallyData.cfg`, `TallyData.db`, and `TallyData.log` files. I was not able to find any description of how to archive all election data, including audit logs, in any of the Hart operator's manuals. I view this as a weakness of the Hart documentation.

In contrast, the California Use Procedures for the Hart system specify that election administrators should back up the BOSS, Ballot Now, Tally, Rally, and Servo databases after the election, not just the Tally database. I was not able to determine whether this is sufficient to ensure that all audit log files are saved, but it appears possible that this might be sufficient.

I evaluated what events are logged by various components of the Hart system:

- The Servo functional specification identifies all log entries that can be generated by the Servo component, and provides a table that maps event codes to a short description and more information about the event. It also provides a brief summary of the events that are logged by the eSlate and eScan. This is a positive aspect of the Hart documentation.

  It appears from the system documentation that the Servo component generates an audit log entry each time the firmware on an eScan, eSlate, or JBC is updated from Servo. This is a positive aspect of the Hart system.

- The eSlate with VVPAT printer (known as the VBO unit) generates audit log entries in a number of conditions, including error conditions. It records an audit log entry any time that the voter accepts a VVPAT record, the voter rejects a VVPAT record, the printer fails, the printer experiences a paper jam, or the paper runs out. It records an audit log entry any time that a voter's ballot is cancelled or recorded. Audit log entries are generated in a number of other situations as well. I consider the broad coverage of many relevant events to be a positive aspect of the eSlate component.

- The eScan generates audit log entries for a number of conditions, including every time that a ballot is cast, a ballot is rejected, or a rejected ballot is overridden and accepted. Audit log entries are generated in a number of other situations as well. The eScan functional specification identifies all log entries that can be generated by the eScan, and provides a table that maps event codes to a short description and more information about the event. I consider the broad coverage of many relevant events a positive aspect of the eScan component.

- The JBC functional specification does not document the audit log entries that are generated. It mentions that the JBC generates a log entry when a ballot is cancelled, but otherwise refers to the PVS Audit Log specification document, which does not appear to be contained in the TDP.

- The system documentation for the eCM Manager, which is responsible for cryptographic key management and generating new cryptographic tokens, suggests that the eCM Manager does not generate any audit logs. The creation of a cryptographic token is a significant event that can affect system security. Therefore, in my view, this is a weakness in the system.

Unfortunately the TDP documents that describe the Hart audit logs in detail are marked "Confidential." This is a weakness of the system documentation.

Several of the documents I reviewed reference "PVS Audit Log Specification Document, Part No. 6000-011." This was not included as part of the TDP. This is a weakness of the system documentation.

The Hart document review report produced as part of the California Top-To-Bottom Review also raised two other concerns:

- The document review report states that the operator's manuals do not clearly explain how election administrators should use audit logs [23, §5.5.2.2]. This matches my assessment of the Hart operator's manuals. Also, the system does not appear to provide the capability to analyze the audit logs and produce a summary report highlighting important action items for administrators. A potential consequence of these weaknesses is that election administrators may not find much value in the audit logs under ordinary operating conditions.

- The document review report raises concerns about whether the system adequately records information about manual adjustments to vote totals. The Hart system allows election administrators to manually adjust vote totals or the number of ballots cast. While the Tally audit logs do record this event, the document review report raised concerns about whether the audit logs include sufficient information to fully determine what adjustments were made [23, §5.5.3.2.2]. I was not able to make an independent assessment of this issue from the information available to me.

During the California Top-To-Bottom Review, the Hart source code review report identified a weakness in the protection of voter privacy. The report found that the eSlate audit log contains information that could violate voter privacy and could be used to correlate voters to their votes, under certain conditions [26, Issue 25, §7.1]. As far as I can tell, this concern remains relevant today.

The Ohio EVEREST study reported that it is possible to take actions and then erase any trace of those actions from the audit log. The trick is to save the Tally database, perform the action, then restore the database—which restores the audit log to its state at the time the database was saved [37, §18.3.4]. It is not clear to me whether this feature is desirable. The EVEREST study also reported that an authorized user of an election management component could delete or modify audit logs generated by that component, if the user is malicious and knowledgeable [37, §20.1.4].

4.5 LA County MTS

Summary. Because only limited information about the MTS system was available to me, and because the vendor responses to questions from the California Secretary of State's office were lacking in detail, I was not able to draw conclusions about the extent to which the MTS voting system's audit logs meet the criteria for audit logs outlined earlier, nor was I able to draw any conclusions about the strengths or weaknesses of the MTS audit logs.

Details. The vendor did not respond to the following questions in the California Secretary of State's questionnaire to vendors:

> c) Are the following events logged in "a concrete, indestructible archival record of all system activity related to the vote tally"? (1990 VSS, §4.8; 2002 VSS, §2.2.5.1; 2005 VVSG, §2.1.5) If yes, in which log(s) are they maintained?
>
> d) Is it possible to delete or modify any audit log entry or entries?
> i) If so, which log(s)? How can the entry or entries be deleted or modified, and by whom?
>
> h) i) Please provide electronic copies of any manuals or other written materials that are available to technicians and programmers concerning the characteristics and use of audit logs.

ii) Alternatively, please give the location of such manuals or other written materials in any technical data package(s) previously submitted to the California Secretary of State.

The vendor did not identify which events are logged or whether the MTS voting system meets the requirements in the federal standards regarding which events must be logged. The vendor's responses to several other questions were cursory, vague, and non-specific. For this reason, I was not able to draw any conclusions about the extent to which the MTS system's audit logs meet the criteria for audit logs outlined earlier, or about the strengths and weaknesses of the MTS system's audit logs.

The MTS system has never been submitted for review under the federal voting standards and I do not know whether it complies with the requirements governing audit logs found in the federal standards.

4.6 Premier GEMS

Summary of strengths. All major system components generate audit logs. The operator's manuals provide a relatively clear description for election administrators of how to perform several routine tasks. The manuals incorporate review of the audit logs as a routine part of the official canvass under certain conditions.

Summary of weaknesses and concerns. The system does not automatically collect audit logs from AV-TSX and AV-OS units in the field, as part of normal operations. The system does not appear to provide any way to export all audit logs to an electronic format suitable for archival and/or automatic analysis by third parties. The system's design makes collecting, reviewing, or sharing audit logs likely to be a tedious and time-consuming task. The system does not provide functionality to analyze or summarize the audit logs.

Detailed analysis. The GEMS User's Guide describes how to print audit logs produced by the election management components and the AV-OS central scanner after the close of elections (§12.6). The Guide provides clear explanations about how to perform this task. However, it does not describe how to archive those logs in electronic format or export them to a format suitable for automatic analysis. It does recommend backing up the GEMS database, which contains these logs in electronic format. The User's Guide does not discuss saving or printing audit logs produced by the AV-TSX units or AV-OS precinct scanners.

The GEMS Election Administrator Guide does provide recommendations for archival that involve printing the audit logs produced by every AV-TSX and AV-OS unit. However, due to shortcomings of the system, its recommended procedure is onerous and burdensome. The Administrator Guide recommends that after every election, election administrators should find every AV-OS and AV-TSX memory card, insert each memory card into an AV-OS/AV-TSX unit, and use the unit to print a copy of the audit log for that memory card, one by one. The Guide provides a clear explanation of how to perform this task. However, this process is likely to be laborious and time-consuming.

The GEMS Election Administrator Guide recommends printing and reviewing the audit log for every precinct where discrepancies are detected during the official canvass. This is good: it increases the chances that audit logs will be examined as part of routine election operations, and thereby increases the chances the some kinds of problems will be detected and corrected. Unfortunately, printing and reviewing the audit log for a particular precinct requires finding the memory card for that precinct, inserting it into the corresponding AV-OS/AV-TSX unit, and using the unit to print a copy of the audit log. This must be performed separately for each such precinct and thus has the potential to be time-consuming.

The system does not provide support for automatically collecting all audit logs into a single location. It would be better if audit logs from every unit were automatically uploaded to GEMS, as part of the process of uploading the votes from each unit, so that all AV-OS/AV-TSX audit logs are available on GEMS at the click of a mouse. The design of the system makes collecting audit logs, cross-correlating them, reviewing them, and archiving them more onerous than necessary. In my view, this is a weakness of the system.

The system documentation on the content of audit logs was generally sparse and did not provide comprehensive documentation of what events are logged and how the audit logs operate. As a result of the incomplete and somewhat non-specific documentation, had I been unaware of the issues discovered in the investigation of the Humboldt County failure [11], I would not have been able to identify those issues from the system documentation. I was not able to determine from the system documentation available to me whether those issues have been remedied in more recent versions of the system. One of the issues discovered after the Humboldt County failure is that GEMS does not log deletion of decks of scanned ballots. I was not able to determine with any confidence whether there are other important system events that are also not logged but should be.

GEMS continues to suffer from the systematic weaknesses identified in the aftermath of the Humboldt County failure. The content of the audit logs remains "cryptic and obscure"; their format is not specified in any public document; log entries are split across multiple files, which do not share a common file format; and the system does not provide support for collecting these log files, collating them into a shared format, or analyzing them.

The AV-OS precinct optical scanner appears to record certain statistics about the election, as well as logging certain actions taken by poll workers. The AV-OS does not appear to generate an audit log entry for each ballot cast, though it does log an event each time that the poll worker overrides the machine to force it to accept a ballot that would otherwise be rejected [3]. The AV-OS does not appear to log jammed ballots, which apparently means that certain kinds of ballot stuffing attacks will not appear in the audit log [37, §14.4.1]. The lack of logging for each ballot would appear to be a weakness of the system. As far as I can tell, the AV-OS audit logs do not appear to be automatically uploaded to GEMS, as part of normal operation of the system. The Ohio EVEREST report identified a potential weakness that the AV-OS audit log only has capacity for 512 log entries; after that, old log entries are overwritten and permanently lost [37, §13.2.4]. A mitigating factor is that this shortcoming only occurs when an unusual number of audit log entries are generated [3].

Researchers at the University of Connecticut discovered several minor flaws in the AV-OS audit logs. They discovered that poll workers can print a totals report in the middle of the day, revealing the tally so far. Such improper behavior could potentially compromise ballot secrecy and give political operatives advance information on election trends. The researchers also discovered that, due to a flaw in the design of the AV-OS software, it is possible to print a totals report mid-day in a way that leaves no record in the audit log that the totals report was printed inappropriately [3, §4.2]. The AV-OS ought to log such an event, but it does not. The researchers also discovered that timestamps in the AV-OS audit log will be inaccurate if the AV-OS is left powered on for more than 24 hours [3, §4.3].

The AV-OS central optical scanner appears to log entries to the GEMS Central Count Server log, under the following six conditions: a new deck is created, a deck of ballots has been fully scanned and is committed to the GEMS database, an attempt to scan a deck of ballots has failed due to a loss of network connectivity, a network connection is established, the network connection has been closed, or an internal error occurs. These audit logs are automatically uploaded to GEMS.

One advantage of the AV-TSX audit logs is that, if I understand correctly, the AV-TSX appears to store audit logs from past elections for as long as there as space available in the internal storage. This is beneficial because it appears to mean that accidental deletion or clearing of logs is a recoverable error. As far as I can tell, the AV-TSX audit logs do not appear to be automatically uploaded to GEMS, as part of normal operation of the system.

The Election Media Processor component maintains an error log, which records any failure to write data to a memory card. This log may be viewed by the operator upon request. It is not clear to me whether errors are automatically displayed to the operator or whether the operator must specifically check the error log. It is not clear to me whether this log is uploaded to GEMS or where it is stored on the filesystem.

Premier claimed that GEMS and AV-TSX audit logs cannot be deleted or modified except by deleting

the entire election database. This claim was not substantiated and I consider it questionable, for the reasons outlined in Section 4.1. In Premier's response to questions from the California Secretary of State's office, Premier wrote:

> *CA SOS:* Does this mean that the attack vectors of going through the .mdb file using MSAccess to delete or modify log entries has been addressed?
>
> *Premier:* If the systems are configured according to our documentation then this has been addressed.

Premier did not substantiate or explain this claim. While it would be possible to address the specific attack vector of using an already-installed version of MSAccess, I was not able to determine how and whether the Premier system would prevent other variations of this attack. The source code analysis report from the California Top-To-Bottom Review specifically claims that this attack vector can be exploited even if MSAccess is not available [9, Issues 5.3.2, 5.3.3]. A number of related concerns about the potential for tampering with GEMS audit logs were raised by the red team as part of the California Top-To-Bottom Review [1, §III.1.a–c].

The red team report from the California Top-To-Bottom Review warned that certain actions can be performed from within GEMS without being logged in the GEMS audit logs [1, §III.1.c]. I was unable to perform an independent assessment of this issue as part of this study. The source code report from the California Top-To-Bottom Review stated that the AV-OS audit logs are not protected against malicious tampering [9, Issue 5.1.4]. The California Top-To-Bottom Review document review report suggests that at least one of the testing labs responsible for federal certification of the Premier system did not test the AV-TSX or AV-OS audit logs to determine whether their integrity is protected against improper manipulation, and that the other testing labs' report was so perfunctory as to prevent any evaluation of the testing performed performed [24, §4.1.1, 4.1.12].

The California Top-To-Bottom Review document review report identified a weakness in the system documentation [24, p.40]. The report states that the operator's manuals do not adequately explain how to use the audit logs produced by the Premier voting system:

> Audit Log information: These logs offer substantial value to security and accuracy values, but unless the documentation explains how to use them and the various purposes for which they are relevant, their presence is nearly meaningless. Their value extends from discerning some types of tampering, to identifying system failures or problems and operator errors, to trouble-shooting and improving election operational performance.

The document review report also expressed a concern that, unless the system is configured carefully, the configuration settings may limit the value of audit logs. The document reviewers reported that a system configured for them by vendor representatives had configuration problems that affected the audit logs:

> we performed a configuration audit of the GEMS server that was provided to the TTBR red team testing room at the SOS facility in Sacramento. We were informed by Diebold technical personnel that the GEMS server was configured just as one would be when delivered to an election jurisdiction in California. Part of the configuration audit focused on the security settings on the TTBR server. [...]
>
> We also found that none of the system-level auditing of security events was enabled, despite the fact that the GEMS configuration described in the GEMS 1.18 Server Administrator's Guide states that the security logging should be enabled and verified to be active. Our examination of other operating-system audit logs found that the Windows Application log, System log and Security log were all configured to retain records of events for only 7 days and that the logs were

limited to 512 KB in size. Maintaining all logs of election audit information for a minimum time period (far greater than 7 days) is a requirement for election security audits and other examinations, as well as federal law. Care should be taken to make sure that California counties that have received GEMS servers configured by Diebold are not being placed in violation of federal or California law due to improperly configured logs on their GEMS servers.

It should be noted that, while it is advisable for election officials to configure security settings on systems in a manner consistent with approved security policies, the GEMS configuration documentation that is provided to customers does not describe the procedure for effecting the proper security settings. The customer documentation's sole mention of such settings is in the GEMS 1.18 System Administrator's Guide, which describes a process to view the logs but none to configure them properly. An industrious election official may undertake to fix relevant settings upon viewing that the settings were not correct, but would do so with no help from the Diebold documentation. The only document that offers any significant coverage of platform security configuration is the GEMS 1.18 Server Administrators Guide, which is not to be circulated outside of Diebold.

Future voting system: Assure 1.2. I reviewed the test plan produced by SysTest labs, the testing lab responsible for testing the next version of Premier's voting system (Assure 1.2) [31]. SysTest's test plan was designed for the purpose of evaluating Assure 1.2 against the 2002 federal voting system standards. The document I reviewed contained only the test plan, but was missing Attachments A–G, which apparently contain elaboration on the test plan and the results of testing so far. That document briefly outlined the plan for testing the Assure 1.2 audit logs [31, p.33]. Based on that brief summary, it appears that the test plan includes checking that an audit log exists and that log entries "meet some minimum standards for information contained and clarity/usability of communications," but do not assess whether the audit logs record all relevant events, the accuracy or completeness of audit log entries, protections against deletion or modification, or other important aspects of voting system audit logs. It is possible that the missing attachments contain more information about SysTest's test plan.

Late in 2008, NIST revoked SysTest's accreditation as a NVLAP-accredited test lab. In response, Premier replaced SysTest with iBeta Quality Assurance, another NVLAP-accredited test lab. The test plan from the iBeta testing lab [25] provides somewhat more detail on how iBeta will evaluate compliance with the federal voting standards' requirements on audit logs. It suggests that security testing will include "Attempts to bypass or defeat voting system security including: [...] modifying data in audit logs, [...]" and that "a software source code review will be executed to confirm that: Audit logs report the date and time of normal and abnormal events" [25, p.35]. It is not clear whether iBeta's testing assessed the accuracy or completeness of audit log entries, whether the audit logs record all relevant events, or whether they protect against accidental deletion or modification.

I emphasize that Assure 1.2 is a future version of Premier's voting system, which was federally certified by the EAC in August 2009 but is currently not approved for use in the State of California.

4.7 Sequoia WinEDS

Summary of strengths. All major system components generate audit logs. There is a way to save an electronic copy of the audit logs from the Edge DRE.

Summary of weaknesses and concerns. As far as I could tell, the system does not appear to automatically upload audit logs from all components (Edge DREs, Insight and Insight Plus precinct scanners, 400C central scanners) to a central location. It appears that the documented procedures for backing up the election

database fail to capture the audit logs from several important system components. As a result, election administrators may routinely fail to archive important audit logs. The system documentation does not clearly describe the format of all audit logs, and it is not clear whether there is any supported way to export audit logs to a file format that is amenable to automated analysis by third parties. The system documentation on audit logs is marked "Proprietary and Confidential." The system does not appear to have the capability to analyze the audit logs and produce summary reports highlighting anomalies and important entries that may be of interest to election officials.

Detailed analysis. The Edge DRE records audit logs that capture certain significant system events. I was not able to determine whether the Edge logs an event each time a ballot is cast or cancelled, or precisely what events are recorded in the Edge's audit log. The Edge functional specification provides a description of how to use the Edge to save an electronic copy of the audit log to the results cartridge for that Edge, how to view the audit log, and how to print it. I did not find a statement whether audit logs generated by the Edge DRE are automatically uploaded to the WinEDS server as part of normal operation.

The WinEDS operator's manuals describe how to archive the election database. They do not mention that this step fails to archive the audit logs from the Insight and InsightPlus precinct scanners, the 400C central scanner, and (possibly) the Edge DRE. They do not describe how to archive those audit logs. In my view, these are weaknesses of the system and documentation.

It appears that audit logs from the Insight and InsightPlus precinct optical scanner are not uploaded to the WinEDS election management system as part of the ordinary operation of the system. Sequoia writes:

> *Sequoia:* The Electronic Log is generated and maintained only within the Insight Pack but any number of copies of the Electronic Log can be printed and stored.

Similarly, it appears that audit logs from the 400C central optical scanner are not uploaded to WinEDS as part of normal operations. As a result, it appears that standard procedures for backing up the WinEDS database do not capture audit logs from the Insight, InsightPlus, or 400C scanners. The failure to automatically archive all logs is a weakness of the voting system software. As Sequoia states in a series of responses to questions from the California Secretary of State's office:

> *CA SOS:* i) Do backups made from server logs actually contain the full logs, or contain links to the log folder location on the server (whose contents may have been erased)?
> (1) e.g. Logs stored directly to the server.
>
> *Sequoia:* Database backups contain the full logs.
>
> *CA SOS:* Your response to Question (e) states, "Database backups contain the full logs." Does this include the logs outside of the database? If so, what actions are necessary to backup the logs outside of the database? Are these standard backups that jurisdictions are required to do? If so, is it in the Use Procedures and where?
>
> *Sequoia:* No, logs outside of the databases require additional operator action. For example, archival of the 400 C logs requires that the log file be copied from the 400 C manually. These additional steps are implied in the Use Procedures, but not specifically stated.

The failure to explicitly identify all steps needed to perform a complete archive of all audit logs (as needed to work around the voting system's failure to do so automatically) is a weakness of the documentation.

The system does not appear to provide any functionality to generate reports that analyze or summary the audit logs.

According to Sequoia's response to questions about what events are logged by the Insight and InsightPlus precinct optical scanners,

Sequoia: Error ballot reading and processing are logged, but the reading and processing of normal error free ballots is not. All operator actions are logged as are all detected error conditions.

The lack of logging of normal ballot casts would appear to be a weakness of the system.

4.8 Summary

It was not possible to perform a meaningful assessment of the BCWin and and MTS systems, due to a failure of the vendors to provide detailed information about their audit logs in response to the Secretary of State's questionnaire.

My review of voting system documentation revealed several weaknesses and limitations of the audit logs produced by the ES&S, Hart, Premier, and Sequoia voting systems:

- All four voting systems record information about some kinds of events to the audit logs. At the same time, there appeared to be opportunities for improvement in this regard. The completeness of the logs varied from system to system; based upon the documentation available to me, it appears that the Hart system produces the most complete audit logs.

- Only the Hart system provides facilities for routinely collecting audit logs from every device after the election. None of the other systems provide good support for collecting log data.

  Even the Hart system has room for improvement in log collection. Depending upon how the Hart system is used, log data may or may not be collected to a central location after the election. For instance, if the Hart system is used with regional vote centers, log data is not collected from all devices and uploaded to the central system.

- None of the systems provide tools for analyzing log data or extracting actionable information from this data. None of the systems provide tools to help election officials make good use of the audit logs.

- All of the systems raise significant barriers to third-party analysis or use of audit log data.

In short, the voting systems' support for audit logs is incomplete: while the voting systems do record some events in their audit logs, they do not provide good support for making good use of these logs.

Chapter 5

Potential future directions and remedial measures

Finally, I outline several potential future directions for improving voting system audit logs or mitigating their weaknesses. I have tried to identify as many potential options for improvement as possible. I am not in a position to assess all of the costs and benefits of these options, or to weigh their relative advantages and disadvantages. The purpose of this chapter is emphatically not to recommend or endorse any particular option, but rather to identify a list of potential directions that the Secretary of State could consider if that is considered appropriate.

5.1 Measures that do not require testing and recertification/reapproval

5.1.1 Option: Do nothing

One possible option is to continue with the status quo, at least for now. My studies have suggested that, at present, audit logs are used or examined only infrequently, so it is not obvious to me whether the benefits of better audit logs would outweigh the costs of improving them. Of course, resources for improving elections are limited, and there are many other worthy ways of using those resources to improve elections; I cannot assess the relative importance of improving audit logs.

5.1.2 Option: Develop guidance for local election officials

Another possibility might be to develop guidance to assist local election officials who use these voting systems. Some of the weaknesses and limitations identified in this report could potentially be mitigated if election officials had clear guidance on how to use, share, and archive audit logs. The Secretary of State could consider issuing guidance for local election officials on two topics:

- **How to archive audit logs.** As mentioned in Chapter 4, in many voting systems, the audit logs are distributed among multiple files. Standard instructions for electronic archival are not always sufficient to ensure that all audit logs are archived. It may be possible to develop a short document for each voting system, describing the steps that local election officials can take to ensure that all audit logs are archived.
- **How to export audit logs for publication.** As identified in this report, in several voting systems, the audit logs may not be in a format that is well-suited to release to members of the public. It might be possible to develop a short document for each voting system, describing how to export audit logs into

a format suitable for release to election observers, candidates, political parties, and others who are interested in analyzing these logs.

For each of these two topics, the Secretary of State's office could consider preparing a short document for each voting system, describing how local election officials can perform these tasks. It might be possible to work together with the voting system vendor to develop such a how-to document. Alternatively, the California Use Procedures for that system could be expanded to provide detailed instructions on how to accomplish these tasks.

5.1.3 Option: Examine directions to support public disclosure of audit logs

Today, audit logs are not routinely shared with members of the public, election observers, and other interested parties. On the other hand, as discussed in this report, broader disclosure of audit logs might facilitate transparency and public oversight of some aspects of elections. It is possible that exploring the barriers to public disclosure of audit logs might lead to new ways to facilitate broader availability of these logs. Some possibilities include:

- **Clarifying the legal status of audit logs.** The Secretary of State could consider analyzing the legal status of voting system audit logs. Are audit logs required under existing law to be released to members of the public, upon request (perhaps in response to a public records request)? Are county election officials permitted to share audit logs with members of the public, if they choose to do so?

- **Clarifying the technical status of audit logs.** The Secretary of State could consider requesting that voting system vendors clarify the status of audit logs and whether they must be kept confidential. For instance, the Secretary of State could ask each voting system vendor whether, in their view, the integrity of an election conducted using their voting system relies upon keeping audit logs secret, and whether there are any specific technical reasons for restricting access to audit logs. If a vendor suggests that the integrity of elections conducted using their voting system may rely upon the secrecy of the audit logs produced by their system, the Secretary of State could consider investigating the extent to which the voting system is able to prevent access to audit logs and any implications this might have on the suitability of the voting system for use in public elections.

- **Clarifying the status of audit log documentation.** As identified in this report, at present most documentation on voting system audit logs has been uniformly marked "proprietary and confidential" by vendors, and as a result cannot be released to the public. There is little technical information available on audit logs that is free of such marking and eligible for public release. It is not clear to me whether there is a solid justification for treating this documentation as confidential. The Secretary of State could ask vendors to clarify whether all of that documentation needs protection from public disclosure, and could inquire whether it would be possible to release much or all of those documents, in the interest of transparency.

- **Collecting audit logs.** The Secretary of State's office could consider routinely collecting voting system audit logs from counties, after every election, similarly to the way the office collects official election tallies after every election. The Secretary of State's office could also consider releasing those audit logs to members of the public or interested parties, on its own authority.

5.1.4 Option: Build collaborations to develop log analysis tools

The Secretary of State could explore building collaborations with third parties to develop software tools for analyzing voting system audit logs. One possible goal would be to develop software tools that would be of

assistance to local election officials. For instance, it seems possible to develop tools that can analyze audit logs and produce summary reports for election officials, such as the ones outlined in Section 3.1.9.

It might be possible to identify non-profit organizations or open-source groups that would be able to assist in such an initiative. One possible model might be the collaboration between the Connecticut Secretary of State and the University of Connecticut's VoTeR center. Another possibility might be to work with the VoTeR center to build upon their tools for analyzing audit logs from Premier AV-OS machines. One challenge is that any third-party effort to analyze voting system audit logs might need access to sample audit logs, and possibly to technical documentation from the voting system vendor, which may be difficult to obtain.

5.1.5 Option: Encourage tools for converting logs to an open format

Another possibility might be to develop software tools for converting voting system audit logs to an open format. As identified in this report, at present some voting systems store audit log entries in a proprietary file format. The Secretary of State could explore options for tools that convert logs from this format to an open, documented format that can be shared with the public and is amenable to automated analysis. The Secretary of State could explore working with third parties to develop such tools, or could explore measures that might incentivize vendors to provide such tools with future voting systems.

5.1.6 Option: Require vendors to document audit log features

Another possibility would be to require vendors to document the extent to which their voting systems meet an expanded list of criteria for audit logs. The voting system standards that current voting systems have been certified against have fairly limited requirements on voting audit logs, and these requirements are not always clear. The Secretary of State could designate a more thorough list of criteria and requirements for voting system audit logs, and then require vendors to identify the extent to which they comply with these criteria. Vendors could be required to state, for each criterion, whether that criterion is or is not met by their voting system; for criterion that are not met, the vendor could be asked to explain why that requirement was not met and assess the impact of failing to meet that requirement. The results of this self-assessment could be taken into account by the Secretary of State during testing and approval events, published by the Secretary of State, and taken into account by counties during the procurement process. It is possible that greater transparency on the strengths and weaknesses of voting system audit logs might encourage development of future systems with improved audit logs, and might enable vendors to compete partially upon the quality of their audit logs.

There are a number of more comprehensive sources for desirable attributes of voting system audit logs. The Proposed VVSG 1.1 contains a slightly extended list of requirements for voting system audit logs. The TGDC's Recommended VVSG 2.0 contains a more comprehensive and rigorous suite of requirements. Chapter 3 of this report contains a detailed list of criteria and desirable features of voting system audit logs.

This requirement would place the burden of assessing compliance upon vendors, rather than upon federal testing labs or state voting system examiners. A potential advantage of having vendors self-identify the extent to which their systems meet these requirements is that vendors are likely to be well informed of the properties of their systems, so this approach might place less regulatory burden on vendors than independent testing. A potential disadvantage is that vendors are not a neutral, independent party, and this might affect the thoroughness or perceived thoroughness of these self-assessments. If desired, it might be possible for state voting system examiners to spot-check a random sample of the vendor's claims, as an independent check on the validity of the vendor self-assessment.

5.2 Measures that require testing and recertification/reapproval

5.2.1 Option: Consider evaluating audit logs as part of the state approval process

One way to encourage improvements to voting system audit logs might be to expand the state approval process to include an evaluation of the voting system's audit logs. The Secretary of State could consider incorporating testing of audit logs into the tests performed by state examiners. For instance, if the state conducts a mock election using the voting system, after the mock election the examiners could perform the following additional checks:

1. Check whether the system is able to produce reports summarizing the audit log contents, and if so, assess how useful the reports are.

2. Collect all audit logs and test the process of sharing them with a third party. Check to see if this process is well-documented. Check whether the resulting log entries are understandable, documented fully in publicly available specification documents, and stored in an open format amenable to automated analysis.

3. Test the process of archiving all audit logs after the election. Check whether this process is well-documented and whether the voting system documentation clearly identifies all steps necessary to store all system logs.

4. Check whether the voting system documentation clearly identifies any steps necessary to configure operating systems, databases, and other third-party software so they will record all events that may be useful for post-election auditing.

The Secretary of State could consider assessing the strengths and weaknesses of each voting system's audit logs, when the voting system is considered for approval in the State of California. Such an assessment might inform the Secretary of State's decision whether to approve the voting system, and might also inform counties' procurement decisions as they buy voting systems.

This additional testing would require more time and resources from state voting system examiners, so the potential benefits would need to be weighed and balanced against the costs.

5.2.2 Option: Consider ways to encourage future voting systems with improved audit logs

The Secretary of State could consider ways to encourage voting system vendors to provide better audit logs in next-generation voting systems. It would be technically feasible, within the state-of-the-art, to address many of the weaknesses or limitations identified in this report. However, it is not clear how to build a market or regulatory structure to incentivize and facilitate such improvements to voting system audit logs.

It might be possible to facilitate a conversation between county election officials, technical experts, voting system vendors, and others. County election officials might be able to explain what functionality they would find useful; technical experts might be able to offer ideas on what functionality could feasibly be provided by a voting system; and voting system vendors might be able to offer information about the likely costs of providing new functionality. One possibility might be to hold a workshop on the topic, inviting election officials, vendors, researchers, election observers, political parties, criminal investigators, and users of audit logs, to facilitate a broader sharing of their perspectives.

Researchers have recently developed innovative new methods for improving the utility of audit logs [17, 18, 33, 36, 38, 3]. These new methods provide the capability to record audit logs redundantly with integrity and privacy, the capability to analyze audit logs for anomalies and other relevant information, and the capability to record a "movie" of each voter's interaction with the voting system that can be replayed

by investigators. This research might be able to provide a basis for improved logging in future-generation voting systems. However, it is not clear how these advances might translate to voting system standards and testing processes. One possible option is that the Secretary of State could suggest that the EAC, NIST, or the TGDC assess the future of audit logs and consider how voting system standards could be updated.

Another possible option is that the Secretary of State could consider requiring that new voting systems submitted to the state for approval comply with stricter requirements on audit logs. The Proposed VVSG 1.1 and the TGDC's Recommended VVSG 2.0 are two possible sources of stricter requirements. A more aggressive option would be to require new voting systems submitted for approval to comply with all the requirements on audit logs found in the TGDC's Recommended VVSG 2.0. One challenge would be to identify ways to assess compliance with these stronger requirements, without introducing undue regulatory burden on voting system vendors or unduly increasing barriers to entry for new vendors interested in introducing innovative new voting systems. A less aggressive option would be to require new voting systems submitted for approval to document the extent to which they comply with these requirements, and for each requirement that is not met, explain why that requirement was not met and the impact of failing to meet that requirement. (See also Section 5.1.6.) The Secretary of State would need to evaluate the costs and benefits of these options.

5.3 Third-party applications

5.3.1 Third-party applications to supplement what is logged in real time

As identified in this report, central election management systems generally do not maintain comprehensive audit logs of all relevant events that occur and all actions taken by election workers. It might be possible to use third-party applications, developed by someone other than the voting system vendor, to mitigate this weakness.

The most promising place where this approach could be applied is logging of events that occur on the central election management system, as opposed to events that occur on polling-place devices. It would be technologically feasible for a third-party application to log additional information about events and actions that occur on the election management system. For instance, it might be possible to log every change to the election database or every change to the filesystem. It might also be possible to log every action taken by the user of the election management system, perhaps by logging every keystroke typed, every mouse click, and a screenshot of every screen displayed to the user. This would in effect record a movie of everything done on the election management computer, analogous to the way a security camera records all activity at a particular location. It might also be possible for a third-party application to automatically configure the operating system and database to increase the amount of information they log. For instance, it would probably be possible to configure the operating system to log an event whenever a user logs in or logs out of the system, starts or exits a software program, or attaches a removable storage medium.

There would be some technical challenges in adopting third-party applications to supplement voting system audit logs:

- **Compatibility and reliability.** The most significant challenge is to ensure that the third-party application does not negatively impact the operation of the voting system software. It is critical to avoid any kind of conflicts that might interfere with the operation of the voting system. For this reason, it will likely be necessary to test the third-party application together with the election management software, to ensure that it does not cause conflicts or incompatibility. Configuring the operating system or database to increase the amount of information they log would be a relatively low-risk change that might not require detailed testing. Third-party software that works by modifying or "hooking"

the operating system or voting system software would be a higher-risk change that could benefit from detailed testing.

- **Storage.** Logs generated by third-party applications might be voluminous and might require extra space. Therefore, it would be prudent to estimate the amount of hard disk space required to support third-party logging software, before installing that software. It is possible that election management servers might need to be re-configured with additional hard disk space, to accomodate the third-party logs. I do not expect this to be a barrier or major problem; I would expect that these issues could likely be addressed without much difficulty, simply by buying additional storage.

- **Security.** Third-party logs should be designed to avoid storing information that might compromise the security of the voting system. For instance, a third-party logging application should avoid logging passwords entered by system operators and election workers, because disclosure of those passwords could negatively impact system security. This might require the third-party application to be specially configured or specially designed with the voting application in mind. I would expect that these goals could be met with relatively modest software development, but they might make it difficult to use off-the-shelf third-party software that was not written specifically for use in elections.

I did not investigate whether there are any third-party software products commercially available that perform this kind of additional logging. There is a high-level overview of commercial offerings on log management and Security Information and Event Management (SIEM) in the NIST guide on log management [30].

I would not expect third-party applications to be effective at supplementing what is logged by polling-place equipment, such as optical scanners, DREs, and voter card activators. Polling-place devices are generally designed as custom single-purpose embedded devices. These devices are normally not designed to support third-party applications, and I would expect that running third-party software on them might raise a number of compatibility, certification/approval, and reliability concerns.

5.3.2 Third-party applications for log analysis

As identified in this report, existing voting systems do not provide tools to analyze audit logs and produce actionable information for election officials, observers, and others. Third-party applications for log analysis are a promising way to address this shortcoming. Log analysis seems like a particularly appropriate task where third-party software could help, because log analysis does not require tight integration with existing voting system software, and because log analysis software does not require federal certification or state approval.

At present, I am not aware of any requirement for third-party log analysis software to be federally or state approved. This seems appropriate to me. I do not see any reason to require that third-party log analysis software undergo certification, testing by federal testing laboratories, or approval by state or federal authorities. I anticipate that election workers would export audit log data from their certified and approved voting system, then import that data into log analysis software running on a separate machine. So long as the log analysis software does not modify the voting system itself, and is run separately, it does not seem to pose any risks to the election. For this reason, log analysis seems like a prime candidate for third-party software.

One challenge is that third-party log analysis software can only analyze whatever events are recorded by the voting system. As identified in this report, existing voting system audit logs are incomplete and do not record all events that might be of interest. Third-party log analysis software would not be able to do anything about this shortcoming; it could only analyze whatever log entries are produced by the existing

voting system. Omissions in the audit logs produced by existing voting systems would likely lead to "blind spots" that no amount of post-facto analysis could eliminate.

Another challenge is that many existing voting systems do not provide a convenient way for election workers to collect audit logs from all polling-place devices and export them for third-party analysis. In these systems, the task of collecting all audit logs from all devices is likely to be tedious and labor-intensive. For this reason, it might be easier to analyze audit logs from central election management systems than from polling-place devices, simply because it would be easier to collect the audit logs from central systems than to collect the audit logs from polling-place devices.

I am not aware of any existing off-the-shelf third-party applications for analysis of voting system logs. The University of Connecticut's VoTeR center has apparently developed some tools for analysis of logs from Premier AV-OS scanners, in collaboration with the Connecticut Secretary of State. I also understand that several computer scientists have developed their own private tools for analysis of ES&S iVotronic event logs. However, in both cases, these tools are limited in scope and it is not clear that they would meet the needs of election officials in their current form. For this reason, I expect that new software for log analysis would need to be developed; no existing software is likely to be adequate in its current form.

A third challenge is that developing third-party software for log analysis may be difficult without detailed technical information about the format of logs produced by existing voting systems. As identified in this report, the voting systems I studied store audit logs in proprietary file formats and do not appear to provide a convenient way to export their audit logs in an open format suitable for third-party analysis. No voting system provided publicly available documentation on the format of log files or the meaning of data in those files. In some cases, the Technical Data Packages (TDPs) contain partial information on file formats, but that information is typically marked Proprietary and Confidential and hence is not eligible for release to third-party vendors interested in building log analysis software. If that information was available to third parties interested in developing log analysis software, it might reduce some barriers to development of third-party applications for analyzing voting system logs. At the same time, I do not want to overstate the usefulness of the TDP documents: I found that they do not fully and clearly document the file formats and generally do not document the meaning codes and other data in the file, so while they might provide some hints that might partially assist with the task of reverse-engineering proprietary file formats, third-party software developers may still face non-trivial challenges.

As mentioned in Section 5.1.4, it may be possible to work with non-profit organizations or open-source groups to develop of third-party log analysis software. This direction would need to be explored further with the relevant organizations.

Bibliography

[1] Robert P. Abbott, Mark Davis, Joseph Edmonds, Luke Florer, Elliot Proebstel, Brian Porter, Sujeet Shenoi, and Jacob Stauffer. Diebold red team report, 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/red_diebold.pdf.

[2] Robert P. Abbott, Mark Davis, Joseph Edmonds, Luke Florer, Elliot Proebstel, Brian Porter, Sujeet Shenoi, and Jacob Stauffer. Hart InterCivic Red Team Report, 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/red_hart_final.pdf.

[3] Tigran Antonyan, Seda Davtyan, Sotirios Kentros, Aggelos Kiayias, Laurent Michel, Nicolas Nicolaou, Alexander Russell, and Alexander Shvartsman. Automating voting terminal event log analysis. In *2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections*, 2009.

[4] atsec information security corporation. ES&S Unit 3.0.1.1 Source Code Review, February 2008. http://www.sos.ca.gov/elections/voting_systems/unity_3011_source_code.pdf.

[5] M. Bellare and B. Yee. Forward integrity for secure audit logs. Technical report, UC at San Diego, Dept. of Computer Science and Engineering, November 1997.

[6] Matt Bishop, Sean Peisert, Candice Hoke, Mark Graff, and David Jefferson. E-voting and forensics: Prying open the black box. In *2009 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections*, 2009.

[7] Matt Bishop and David Wagner. Risks of e-voting. *Communications of the ACM*, 50:120, 2007.

[8] Matt Blaze, Arel Cordero, Sophie Engle, Chris Karlof, Naveen Sastry, Micah Sherr, Till Stegers, and Ka-Ping Yee. *Source Code Review of the Sequoia Voting System*. California Secretary of State's "Top to Bottom" Review, July 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/sequoia-source-public-jul26.pdf.

[9] Joseph A. Calandrino, Ariel J. Feldman, J. Alex Halderman, David Wagner, Harlan Yu, and William P. Zeller. *Source Code Review of the Diebold Voting System*. California Secretary of State's "Top to Bottom" Review, July 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/diebold-source-public-jul29.pdf.

[10] Transcript of March 17, 2009, Public Hearing, 2009. http://www.sos.ca.gov/elections/voting_systems/premier/gems11819-hearing-transcript.pdf.

[11] California Secretary of State Debra Bowen. California Secretary of State Debra Bowen's Report to the Election Assistance Commission Concerning Errors and Deficiencies in Diebold/Premier GEMS Version 1.18.19, March 2009. http://www.sos.ca.gov/elections/voting_systems/sos-humboldt-report-to-eac-03-02-09.pdf.

[12] Election Assistance Commission. 2005 Voluntary Voting System Guidelines, December 2005. http://www.eac.gov/program-areas/voting-systems/voluntary-voting-guidelines/2005-vvsg.

[13] Election Assistance Commission. EAC Decision on Request for Interpretation 2009-04 (Audit Log Events), September 2009. http://www.eac.gov/program-areas/voting-systems/docs/eac-decision-on-request-for-interpretation-rfi-2009-04-audit-log-events-final-9-29-09.pdf/attachment_download/file.

[14] Election Assistance Commission. Proposed Draft Revisions to 2005 Voluntary Voting System Guidelines, June 2009. http://www.eac.gov/program-areas/voting-systems/voting-system-certification/2005-vvsg/draft-revisions-to-the-2005-voluntary-voting-system-guidelines-vvsg-v-1-1.

[15] Federal Election Commission. Voting systems standards, 1990.

[16] Technical Guidelines Development Committee. Voluntary Voting System Guidelines Recommendations to the Election Assistance Commission, August 2007. http://www.eac.gov/program-areas/voting-systems/voluntary-voting-guidelines/draft-voluntary-voting-system-guidelines-delivered-to-eac/.

[17] Arel Cordero and David Wagner. Replayable Voting Machine Audit Logs. In *2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT'08)*, August 2008.

[18] Paul T. Cotton, Andrea L. Mascher, and Douglas W. Jones. Recommendations for voting system event log contents and semantics. In *NIST Workshop on a Common Data Formats for Electronic Voting Systems*, October 2009. http://vote.nist.gov/cdf-workshop/mascher-cotton-event-log.pdf.

[19] Federal Election Commission. *Voting System Standards*, 2001. http://fecweb1.fec.gov/pages/vss/vss.html.

[20] Freeman Craft McGregor Group (FCMG) Red Team. Red Team Testing of the ES&S Unity 3.0.1.1 Voting System, February 2008. http://www.sos.ca.gov/elections/voting_systems/unity_3011_red_team.pdf.

[21] Computer Security Group. Security Evaluation of the Sequoia Voting System: Public Report, 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/red_sequoia.pdf.

[22] J. Alex Halderman, Eric Rescorla, Hovav Shacham, and David Wagner. You Go to Elections with the Voting System You Have: Stop-Gap Mitigations for Deployed Voting Systems. In *2008 USENIX/ACCURATE Electronic Voting Technology Workshop (EVT'08)*, August 2008.

[23] Joseph Lorenzo Hall and Laura Quilter. Documentation Review of the Hart InterCivic System 6.2.1 Voting System, July 2007. California Secretary of State's "Top to Bottom" Review.

[24] Candice Hoke and Dave Kettyle. Documentation Assessment of the Diebold Voting Systems, July 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/diebold_doc_final.pdf.

[25] iBeta Quality Assurance. Premier Election Solutions ASSURE 1.2 VSTL Certification Test Plan (Version 2.0), April 2009.

[26] Srinivas Inguva, Eric Rescorla, Hovav Shacham, and Dan S. Wallach. *Source Code Review of the Hart InterCivic Voting System*. California Secretary of State's "Top to Bottom" Review, July 2007. http://www.sos.ca.gov/elections/voting_systems/ttbr/Hart-source-public.pdf.

[27] Douglas W. Jones. Auditing elections. *Commun. ACM*, 47(10):46–50, October 2004. http://www.cs.uiowa.edu/~jones/voting/cacm2004.shtml.

[28] Douglas W. Jones. *The European 2004 Draft E-Voting Standard - Some critical comments*, October 2004. http://www.cs.uiowa.edu/~jones/voting/coe2004.shtml.

[29] Douglas W. Jones. Computer security versus the public's right to know, May 2007. http://www.cs.uiowa.edu/~jones/voting/cfp2007.pdf.

[30] Karen Kent and Murugiah Souppaya. Guide to computer security log management, March 2006. NIST Special Publication 800-92, http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf.

[31] SysTest labs. Election Assistance Commission Voting System Certification Testing: Certification Test Plan, May 2008. Document Number 06-V-DB-058-CTP-01, Rev 09.

[32] Petros Maniatis and Mary Baker. Secure history preservation through timeline entanglement. In *Proceedings of the 11th USENIX Security Symposium*, San Francisco, CA, August 2002.

[33] Andrea L. Mascher, Paul T. Cotton, and Douglas W. Jones. Improving voting system event logs. In *RE-Vote'09: First International Workshop on Requirements Engineering for E-voting Systems*, August 2009.

[34] David Molnar, Tadayoshi Kohno, Naveen Sastry, and David Wagner. Tamper-evident, history-independent, subliminal-free data structures on PROM storage -or- how to store ballots on a voting machine (extended abstract). In *2006 IEEE Symposium on Security and Privacy*, May 2006.

[35] County of Alameda. Application For In Camera Review of Records In Support of Motion For Summary Judgement or To File Records Under Seal In The Alternative; and Memorandum of Points and Authorities In Support Thereof, 2006. Case No. RG04-192053, Alameda County Superior Court.

[36] Sean Peisert, Matt Bishop, and Alec Yasinsac. Vote selling, voter anonymity, and forensic logging of electronic voting machines. In *42nd Hawaii International Conference on System Sciences (HICSS)*, January 2009.

[37] Project EVEREST (Evaluation and Validation of Election-Related Equipment, Standards, and Testing). *Risk Assessment Study of Ohio Voting Systems*, December 2007. `http://www.sos.state.oh.us/sos/info/everest.aspx`.

[38] Daniel R. Sandler and Dan S. Wallach. Casting votes in the Auditorium. In *Proceedings of the 2nd USENIX/ACCURATE Electronic Voting Technology Workshop (EVT '07)*, Boston, MA, August 2007.

[39] Bruce Schneier and J. Kelsey. Cryptographic support for secure logs on untrusted machines. In *USENIX Security Symposium*, pages 53–62, San Antonio, TX, January 1998.

[40] Election Systems & Software. Software specifications: Audit manager, November 2005. Version: 7.3.0.0.

[41] Election Systems & Software. Software specifications: Model 100 precinct tabulator, January 2006. Version: 5.2.0.1.

[42] Dan S. Wallach. Security and Reliability of Webb County's ES&S Voting System and the March '06 Primary Election. Expert Report in *Flores v. Lopez*, `http://accurate-voting.org/wp-content/uploads/2006/09/webb-report2.pdf`, May 2006.

[43] Dan S. Wallach. Testimony for Dr. Dan S. Wallach, Texas House Committee on Elections, June 2008. `http://www.cs.rice.edu/~dwallach/pub/texas-house-elections25june08.pdf`.

[44] Kim Zetter. Voting Machine Audit Logs Raise More Questions about Lost Votes in CA Election, January 2009. `http://www.wired.com/threatlevel/2009/01/diebold-audit-l/`.