
# A CASE STUDY OF REAL-TIME ELECTION FORENSICS

*Strategies For Insuring The Integrity of Electronic Voting Systems.*

**Cyrus J. Walker,**
*Managing Principal,*
*Data Defenders , LLC*

**May, 2011**

## TABLE OF CONTENTS

## ABOUT THE AUTHOR

Cyrus J. Walker is Founder and Managing Principal at Data Defenders, LLC, a business-to-business professional services company providing information assurance, election system auditing, infrastructure management, and software development solutions to local, state, and federal agencies as well to the corporate sector.

Cyrus is an experienced Information Security industry executive with a diverse technology background. He has served, or is currently serving, in several executive-level capacities in the commercial and academic sector of the Information Security and Computer Forensics industry.

Cyrus pioneered the concept of Applied Computer Forensics and related forensic analysis models and processes that are now being used by Data Defenders to aid election jurisdictions in ensuring the operational integrity of the their electronic voting systems. Cyrus has been providing election system auditing solutions to the election jurisdiction marketplace since 2007 and has a diverse technical background including expertise in Data Communications, Information Security, and Computer Forensics.

In addition to his responsibilities at Data Defenders, Cyrus served as a subject matter expert on Computer Incident Response Management in the Cyber Defense Analysis Center at the University of Arkansas funded and directed by the U.S. Department of Homeland Security. He also served as chairman of the Computer Security and Computer Forensics department at the Wilbur Wright College, one of the City Colleges of Chicago.

Cyrus's theoretical work on the topic of Modeling, Analysis and Simulation of Computer and Telecommunications Systems has been published in the IEEE journal (1997). His work on Applied Computer Forensics for Election System Auditing was published in the EVOTE - 5th International Conference on Electronic Voting Conference Journal (Bregenz, Austria - 2012). .Additionally, in 1995 President Bill Clinton recognized him for his groundbreaking work and leadership in the telecommunications industry.

Cyrus holds a Computer Forensics Examiner certification from *the Information Assurance Certification Review Board* and possesses a unique technical background in network architecture and design, disaster recovery, information security, and computer forensics.

## INTRODUCTION

Electronic voting systems consist of the various forms of vote capture devices, the Election Management System (EMS) application software, precinct to Election Central vote transmission devices and modalities, and the accessory and peripheral items needed to connect and power these various components. In the 2005 Voluntary Voting System Guidelines (VVSG), the U.S. Election Assistance Commission (EAC) defines a voting system as:

> *The total combination of mechanical, electromechanical or electronic equipment (including the software, firmware, and documentation required to program, control, and support the equipment) that is used to define ballots; to cast and count votes; to report or display election results; and to maintain and produce any audit trail information; and the practices and associated documentation used to identify system components and versions of such components; to test the system during its development and maintenance; to maintain records of system errors and defects; to determine specific system changes to be made to a system after the initial qualification of the system; and to make available any materials to the voter (such as notices, instructions, forms or paper ballots).*

While the EAC definition is comprehensive, for the purposes of this paper we will focus on the vote capture devices and the EMS software that provides those devices with the election definition and allows for results aggregation and reporting. This paper seeks to provide the reader with a basic understanding of how digital forensics can aid jurisdictions in maintaining the integrity of their election equipment in real time.

Optical scanners are the workhorse vote capture device. They have been for many years. For detailed information to back up this point, please review the detailed research done by Kimball Brace of Election Data Services:
http://www.electiondataservices.com/index.php?option=com_content&view=article&id=22&Itemid=25
.

Although Direct Recording Electronic (DRE) voting machines, often called “touchscreen” voting machines, “ATM style” or simply “electronic voting machines”, made inroads into the technology share of optical scanners starting in the 1980s and accelerating after the 2002 passage of the Help America Vote Act (HAVA), optical scanners have and continue to reign as the most prevalent vote capture technology by the numbers of jurisdictions utilizing them.

Optical scan voting machines work by sensing marks made by a voter in designated target areas. These target areas correspond to choices for candidates and measures on a ballot, which may be single or double-sided.

Older technology optical scanners counted down the columns of a ballot looking for voter marks in a given position based on that position's distance from the top edge of the ballot or a printed "start bar" on the ballot. If the scanner sensed a mark in a given position it added a tally to that position's totals in memory. Later, a results tape could be printed, on which the tallies in memory were married to the candidate and measure names associated with the ballot positions, providing a human readable tabulation of that scanner's election results. More modern optical scanners digitally capture the ballot, triangulating the position of the voter target areas and checking at the pixel level for the presence of a voter mark. This technology allows for more flexible ballot layouts, as well as the ability to determine the percentage marked area of a voter target and ascribe to a range of percentages the actions to be taken by the optical scanner unit. These actions include assessing a mark as no mark if a low percentage of the target is filled, assessing a mark as a valid vote if the percentage filled exceeds some threshold, and assessing a mark as an ambiguous mark if it is in some range of percentage filled that is deemed to fall between a no vote and a tally. This last feature is especially powerful in allowing the voter to better fill in a voter target if it has been filled in but marginally. Optical scanners can give differing results during recounts due to marks falling in the ambiguous range, but features that disallow casting a ballot containing ambiguous marks and technology such as digital scanning systems prevent this issue at recount. Furthermore, optical scanners can be outfitted with imprinters, which allow the ballot to be marked in some manner as it is cast and physically on its way out of the scanner and into the ballot box. Imprinting ensures that the ballot cannot be scanned twice in an election. Imprinting also allows for tying the ballot to the electronic record of the scan, typically a TIFF or other image format, stored in the scanner's memory and reported as part of the election results.

DRE voting machines are often associated with providing polling place technology for voters with accessibility issues, primarily visual impairments. Although DREs in their earliest form have been in service since the 1980s (i.e., the Sequoia AVC Advantage and similar machines manufactured by Danaher Controls), the technology of these devices has advanced from push button switch technology to use of true touchscreens or other voter navigation interfaces. The computing aspects such as processors and memory have advanced, benefitting from the major advancements in the technology of computing devices in general. These advances made it possible for DREs to play back ballot audio, interpret signals from personal assistive technology, and present the ballot visually with zoom and contrast control by the voter; and to do so with such speed and robust quality that the voter does not grow impatient with slow machine response, noticeably slow display, and/or poor audio quality.

DREs have been the focus of security concerns since the early 2000s when their deployed numbers increased after the November 2000 General Election due to market demands and the move away from punchcard voting. Typical concerns are that the machines are subject to tampering, that their firmware was deemed a

trade secret and thus not subject to public verification, and that if tampered no record would exist to determine that the unit had undergone tampering as well as allow for reconstructing the election to confidently declare a winner under these circumstances.

Manufacturers of voting machines responded to these concerns by developing Voter Verifiable Paper Audit Trail (VVPAT) technology for their DRE equipment. A VVPAT allows the voter to see his or her selections printed on (typically) a three to four inch wide cash register styled printout and assess if the machine recorded the voter's selections as desired. VVPAT enabled machines allow the voter to go back and change his or her selections, as many times as allowed by state law regarding how many ballots a voter can receive before being required to cast a ballot in that election. The VVPAT does not allow the voter to remove the paper printout. VVPATs are not a panacea; and there are concerns with their security and especially their accessibility and usability. Researchers in California during California Secretary of State Debra Bowen's Top-To-Bottom Review (TTBR) of Voting Systems (http://www.sos.ca.gov/voting-systems/oversight/top-to-bottom-review.htm) were able to program DREs to display a set of results on the VVPAT while recording a different set of results for tabulation. Groups advocating for the blind have raised concerns regarding access to the VVPAT for blind and visually impaired voters. There has also been some research showing that not all voters review the VVPAT at the end of their voting session; although it is arguable that if only a few voters check the VVPAT for accuracy relative to their selection, its use as a tamper deterrent and detector is not negated.

Although optical scanners have become the favored ballot casting technology of many concerned with election integrity, it too, shares vulnerabilities common to other balloting technologies. Like hand counted paper ballots and punchcard systems, the paper records can be destroyed, lost, or manipulated with relative ease. A pen and ballot access is all that is needed to change the apparent choices made by a voter using paper ballot media. Ballots can be removed from the ballot box after being scanned, sorted to suit individual tastes, then re-scanned to influence precinct totals.

When information is in motion across a network or other media, it is vulnerable to fraudulent and inadvertent changes. Voted ballots and results totals arising from them are no exception. Ballots, both blank and voted and results media (hereafter referred to as Election Night Materials) are transported from the precinct to Election Central for tabulation via a number of methods: pollworker personal vehicles, taxicab, jurisdiction employees (including both election board and law enforcement staff members), and electronic transmission. Each of these methods has vulnerabilities. Transport methods which rely on personal vehicles and taxicabs have some obvious vulnerabilities considering the untrusted entities involved, the opportunity for road traffic delays, loss of the materials, and a number of other possibilities. Electronic transmission has the opportunity for interception and manipulation of the messages. Transmissions are thus encrypted and the

sender/receiver pair authenticated. These vulnerabilities are further mitigated by law and custom deeming Election Night results unofficial and subject to revision during the canvass process and post-election audit.

Election Management Systems are typically server-client pairs deployed at Election Central. These deployments share many vulnerabilities with common Information Technology deployments in workplaces and government offices. Some vulnerabilities can be dismissed when the EMS deployment is never attached to the Internet and maintained as an isolated deployment. The EAC requires server and client hardening procedures as a prerequisite to entering functional testing during federal certification test campaigns. See Voting System Test Plans at http://www.eac.gov/testing_and_certification/voting_systems_under_test.aspx for reference. Vulnerabilities of these deployments can arise and be made manifest by internal (trusted) attackers as well as external attackers. There are numerous references to Windows and other software vulnerabilities. An authoritative source is http://www.us-cert.gov/. It is imperative for the mitigation of these and other vulnerabilities of voting systems, regardless of the technology, that proper election administration is practiced.

## CURRENT STATE OF MITIGATIONS

Mitigations being utilized by States and local election officials can be divided by phase of the voting system lifecycle and its intersections with the election process: pre-installation, installation, pre-election, and post-election.

Pre-election mitigations occur during the certification testing of the voting system. At the federal level the US Election Assistance Commission (EAC) requires a number of items that provide for hardware and software verification after installation and after election events. The Voluntary Voting System Guidelines (VVSG), 2005 edition calls for software verification tools to be provided to the EAC before system certification is granted. See VVSG section 7.4 of the EAC Certification and Testing Program Manual which can be found at http://www.eac.gov/testing_and_certification/manuals_forms_and_notice_of_clarifications.aspx.

These tools are developed by the voting system manufacturer and vetted by both the Voting System Test Laboratory (VSTL) and EAC staff members. Similarly, the manufacturer must provide to the EAC a book of photographs that allow a component level (disassembled) inspection of the voting machines in the Certification effort as well as tools to provide for software verification. The EAC libraries the software in the voting system “as certified”, which offers a firm foundation for subsequent software verification efforts at a jurisdiction using a certified system.

Openly disclosed code is thought to have some security mitigation effects. The arguments here range from those commonly ascribed to open source software relative to a perceived higher degree of security arising from the open source development methodology and philosophy, to arguments surrounding public policy – in particular the transparency of voting system software and the responsibility of manufacturers relative to the proprietary nature of their voting system offerings. This paper is not meant to argue this controversial subject.

At installation, jurisdictions perform some level of customer acceptance test. This testing is performed for a number of reasons, from contract requirements fulfillment to simply ensuring the equipment is functional upon arrival, to a full test emulating an election cycle, including some checks to ensure the equipment firmware and the election management software is configured as certified. One commonly used method for ensuring software and firmware integrity at system installation is to re-load the firmware on every voting machine. The firmware binaries are obtained from a trusted source, usually the VSTL. Similarly, the election management software can be obtained from a trusted source and installed on client and server computers. Some jurisdictions take hash values of the firmware before or, if the voting machine is amenable to this method, after installation on the voting machine. New York State has hash checking codified in election law. New York requires a sample of voting machines in each County fleet be hash checked at prescribed intervals. Colorado builds software during the State certification test under trusted build conditions. The State maintains its own software repository and State staff members install voting system software whenever a County needs it due to maintenance, new equipment acquisition, or any other reason. The State of New Jersey is in the process of reloading all of the voting machines in the State with a new firmware load from a trusted source, then tamper evident sealing the voting machines. Other States have some sort of program for pre-election and/or post-election voting system hash checks.

Hash checking the voting system is a good start in attempting to ascertain that the software in place is the software expected. It has some limitations. These limitations are driven by the scale of voting system implementations, both at the software level and the system deployment level. At the software level, an election management system code base can be over one million lines of code spread over a number of code modules and standalone programs. Drivers, utilities, and other off-the-shelf software add to the line/module count.

Moving to the tangible, a jurisdiction may have thousands of voting machines in its fleet. The City of Chicago, with approximately 2,600 precincts, has 7,800 voting devices deployed for every General Election. Small Counties may have few staff members, such that even a few voting machines overwhelm their ability to hash check a statistically significant portion of their fleet. The amount of time required to take hashes is not trivial; and some jurisdictions have far more than the typical two to three elections per year.

Further complicating hash checking, especially the election management software is the customization each installation undergoes to comply with State law and individual jurisdiction needs. Election management software has a significant configurable element due to differences in State laws across the United States. Ranked Choice Voting, crossover Primaries, and Straight Party voting all contribute to differing configurations that may result in different hash values from one jurisdiction to another, especially jurisdictions in different States. Drivers for printers, memory card readers, and other peripherals further complicate taking hashes and arriving at values which can then be defined as "correct" or "incorrect".

Furthermore, election management software contains files that change with every election. Some are completely new for every election. When attempting to hash a directory in the election management server or client, the presence of these files invalidates the known good hash values obtained from the EAC or other election authorities. In fact, files change as the election event proceeds from initial definition through final canvass, making the task of deeming an installation pristine or tampered very difficult.

## WHY COMPUTER FORENSICS IS BETTER THAN TRADITIONAL ELECTION EXAMINATION METHODS

The following chart highlights the pros and cons of each of the integrity management methods highlighted in previous sections of this paper:

| Integrity Management Methods | | |
|---|---|---|
| **Method** | **Pro** | **Con** |
| **Hash Testing** | ***Pre-Use*** software can be validated using proven algorithms. | ***Post-Use*** dynamic files cannot be validated over time. |
| | | It does not validate the integrity of a device's file structure. |
| | | It cannot reliably identify "foreign" code hidden in a system's file structure. |
| **Trusted Build** | ***Pre-Use*** software can be validated using proven algorithms. | ***Post-Use*** dynamic files cannot be validated over time. |
| | Can identify "foreign" code hidden in a system's file structure. | It cannot verify if foreign code is of a malicious nature. |
| **Forensic Analysis** | ***Pre-Use*** software can be validated using proven algorithms. | Forensic analysis requires a huge amount of time to complete in order to be useful to the election management process. |

| | | |
|---|---|---|
| | ***Post-Use*** dynamic files can be validated over time. | |
| | Can identify "foreign" code hidden in a system's file structure. | |

## HASH TESTING METHOD

The premise behind hash testing is that an algorithm is used to generate a unique, mathematical value directly associated to a specific set of data. Depending of the algorithm used, there is an extremely low probability that this value could be reproduced for a different set of data. There are a number of algorithms used such as SHA1 and MD5 that can generate unique hash values.

All systems generally contain two types of files: static and dynamic. Static files are those files where the contents do not change as a result of routine system operations. Dynamic files are those files where the contents do change as a result of routine system operations. The core premise of the hash test method is to compare the pre-use and post-use file state hash values of a particular file in question. The theory is that the hash values of the pre-use and post-use file states are compared to each other to determine if the file's contents have been changed from state to state during routine systems operations. If the hash values of the pre-use and post-use file states match, then it is confirmed that the contents of the file did not change during routine systems operations. If they don't match, then it is confirmed that the contents of the file did change.

The hash testing method is an easy and quick way to confirm whether a file changed or not. The hash values of the pre-use and post-use files states should always match after a static file has been used in some routine system operations because static files are designed to stay static during routine system operations. The hash values of the pre-use and post-use file states should not match after a dynamic file has been used in some routine system operations because dynamic files are designed to change during routine system operation. While the hash testing can quickly confirm whether a file changed or not, this method does not indicate what those specific changes were, nor can it determine whether changes made to a dynamic file were of a malicious nature or not. That requires a secondary, more detailed analysis. This is the major shortcoming of the hash testing method.

Viewing these limitations through the lens of a jurisdiction, the hash testing method is useful for confirming that a jurisdiction has implemented an initially "clean" software build that was delivered to them. Assuming the known good hash values arise from a trusted source such as the US EAC, the jurisdiction can be confident that the software received matches the software certified. Once the software is installed and used, the hash testing method cannot confirm that the software remains clean during system operations- a serious limitation of the hash testing method. A person with malicious intent typically has two overall objectives: to maximize the impact of malicious activities and to minimize his or her exposure during and after the commission of those activities. To accomplish these two objectives when attacking a system being monitored - especially an electronic voting system - the first and best attack vector is to hide malicious code in dynamic files. Why? Because dynamic files naturally change during routine system operation and therefore, as a general matter of course, one can't determine using the hash testing method whether those changes were of a malicious nature or not.

If an attacker were to swap out a clean dynamic file for one that has been maliciously altered, the malicious version of the dynamic file would not be detected by the hash testing method because the hash values of the clean version - or pre-use dynamic file - would naturally be different from the post-use malicious version.

See Figure 1. Hash Testing Method Illustration for reference.

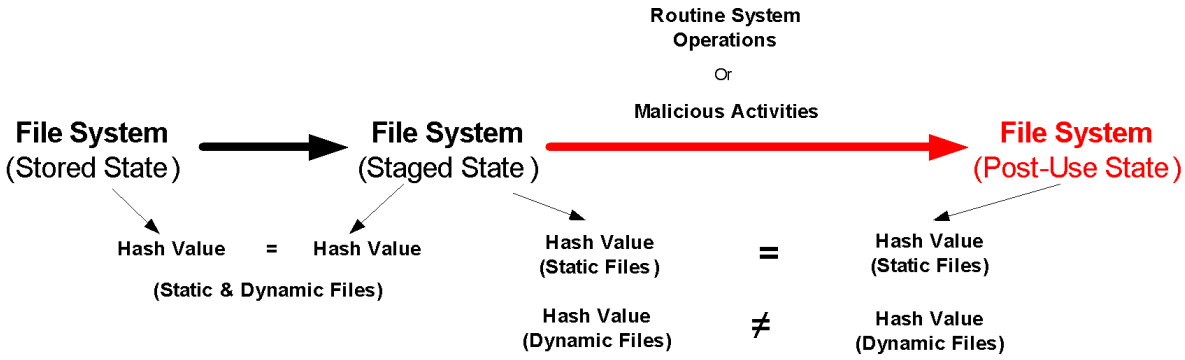


**Figure 1. Hash Testing Method Illustration.**

Additionally, shortcomings of the hash testing method are that it cannot identify the presence of malicious code on a system even if the malicious file made it into the initial build of the software. This method will simply provide hashes for pre and post states of malicious files that would be accepted by those building the trusted build because the intention of those malicious files would not be obviously detectable.

### Trusted Build Method

The premise behind the trusted build method is somewhat the same as the hash testing method. In the trusted build method, the assumption is made that when software code is compiled at the source and is delivered to the user, it is considered clean. Just before being delivery to the end-user, the software build is hashed by the source. That trusted build is stored securely until it is time for end-user deployment. Just prior to deployment, the build is hashed by the end-user and those hash value(s) are compared to the hash value(s) generated by the source. Both hash value sets should match which would then indicate the trusted build received from the source is the same build being deployed at the end-user and therefore, it is considered a "trusted build".

The trusted build method suffers the same shortcomings as the hash testing method in that if you were to hash the software after use in a system, the hash value sets naturally would not match, and therefore, you cannot verify that the software running on the system is still the trusted build. This is based on the consideration that the software is being used in a system/environment that is subjected to other unmitigated attack vectors that could subject the software to the risks of being impacted in a negative manner.

As in the hash test method, once a dynamic file has been used, its post-use file state hash value will naturally be different from its pre-use file state hash value. Therefore, you can't confirm that the software in use is still the trusted build.

Depending on how the trusted build method was executed, you could detect the presence of foreign files that were not in the initial, trusted build, but you could not determine if those files were of a malicious nature without a detailed analysis of those files.

### Forensic Analysis Method

This brings us back to the question, why is computer forensics better than other traditional election analysis methods? To answer this question we will start by providing a brief description of computer forensics.

Computer Forensics refers to the practice of acquiring digital information (from computers and digital storage media) used to determine the root cause of a computer-related event. A computer related "event" means that something occurred that caused routine system operations to be negatively impacted which produced unexpected results. To determine root cause of a computer related event, one must basically determine what happened to the system (what were the results) and how it happened (or what was the cause). Any time the integrity of software or system operations is compromised, this should be considered a computer related event requiring a comprehensive analysis to understand the root cause of the event. Along with the goal of determining root cause, maintaining the verifiable integrity of digital information acquired and analysis processes are also important parts of the computer forensics process.

Computer forensics consists of standards and methods that govern, dictate, and facilitate how data is collected and preserved for use in the analysis process. These standards and methods can be divided into the two main activities of data acquisition and data analysis.

All computers contain some type of "digital artifacts" left behind after an "electronic transaction" has taken place on the computer. These artifacts can be recovered very much like evidence collected at a physical crime scene. Once these digital artifacts are acquired and analyzed, they can provide a wealth of historical information about the event they are associated with. This analysis can lead to an understanding of the root cause of the event which is called the Root Cause Analysis.

Computer forensics process can be used on any electronic device from a computer, cell phone or personal digital assistant (PDA) to the global positioning system (GPS) system in a car, automatic teller machine (ATM), a digital camera, and storage media. If the electronic device stores data, that data can be retrieved and examined using computer forensics. There are few areas today that are not subject to the use of technology. As a result of the growth of the integration of technology in our lives and the increased opportunity for its misuse, computer forensics has become an extremely important process in law enforcement, corporate information technology (IT) management, government IT management, and litigation activities.

The two main activities of data acquisition and data analysis make up the comprehensive analysis mentioned earlier and it is our belief and experience that the forensic analysis method offers the best, most comprehensive solution to mitigate operational and functional based threats against an electronic voting system.

The first activity of data acquisition consists of using specialized processes that make exact duplicates of information collected during the acquisition process. These specialized processes enable the integrity of information collected from devices such as an electronic voting system to be maintained and verifiable at any time. These exact duplicates are often referred to as "bit-stream" copies. A 'bit-stream' copy is an exact, duplicate copy of every bit (1 or 0) written anywhere on the storage media which will include allocated and unallocated space as well as the boot sector of the drive. The bit-stream copy process places each bit found on the suspect drive, in the same physical/logical location on the duplicate drive. In order to maintain the integrity of the data being copied from the suspect drive to the duplicate drive, the bit-stream copy process should be a read-only process from the suspect storage media to prevent any writing to the suspect drive that may overwrite and possibly contaminate or destroy evidence as it is being copied to the duplicate storage media.

There are a couple of reasons for this specialized process. The first being that evidence related to the event could be found in unallocated space or the boot sector of the drive depending on the sophistication of the attack against the system. The objective of computer forensics is to maintain the integrity of digital information as it used during the investigative process. Generating hashes for each file acquired in the investigation process will allow one to validate its integrity. Validating the integrity is extremely important to the outcome of a computer forensics investigation because the results of such an investigation are directly related to the information (evidence) collected for that investigation. The end-goal of a computer forensics investigation it to determine the root cause of a computer related event. The identified root cause has a direct correlation to the evidence. If the evidence was changed in any way during the forensics analysis, or the integrity of the evidence could not be verified, then the identified root cause cannot be confirmed. As a result, the entire investigative process is contaminated and inconclusive. Therefore, this specialized process of acquiring data ensures that the integrity of that data is well maintained which translates into a valid, verifiable root cause analysis in the end.

The second activity of analyzing data (evidence) to determine the root cause of the computer related event is a process requiring a technical understanding of the functionality of the application and/or operating system running the device in question. This investigative function will allow one to identify the reason why the event occurred or answer the question of what was the most basic condition that existed to lead to the cause of the event, the root cause analysis. There are many different investigative techniques that could be used during this process. The basic premise here is to piece together the evidence collected to identify the most logical conclusion for the cause of the event. If for instance, the results of an election for a particular precinct did not match the historical pattern of results for that precinct and it is suspected that the system may have been tampered with in some way, the next step is to confirm tampering by examining the system's file, for example. This could be done by first examining the system to determine the entry point into the system by examining logs generated by the operating system or application.

The computer forensics method can be used to validate pre-use software by using the data acquisition method to acquire and hash the software before it's deployed to a system. For post-use validation, the computer forensics method can easily be used to validate all static file in the system. Post-use validation of the integrity of dynamic files requires a more specific application of computer forensics to track and validate changes made those types of files. Additionally, foreign code hidden anywhere in the system can be identified by using this specific application of computer forensics.

The only drawback to the traditional computer forensics method is that it can be extremely time consuming and resource intensive because large volumes of data can be acquired resulting in the examination of hundreds of thousands, if not millions of files, requiring weeks or months to analyze.

However, this problem is solved in the specific application referred to Applied Computer Forensics discussed in the next section.

## Applied Computer Forensics

What does Applied Computer Forensics™ mean? Applied Computer Forensics (ACF) is the application of the traditional computer forensics process, described in the previous section, in a manner that works in response to a specific need.

The focus of traditional computer forensic is to find the root cause of a computer related event. This positions traditional computer forensics as a purely reactive process. Applied computer forensics provides for a proactive approach to managing the operational integrity of an electronic voting system by identifying changes made to a system that could threaten it.

The application here would be in an election jurisdiction and the need here would be to quickly detect and mitigate any attack against and election system before that attack has a negative impact on the operations or outcome of that election. With, the sheer volume of data acquired from an electronic voting system, what is needed is a way in which to analyze that data quickly to allow the election jurisdiction to be proactive in its response to any impending threat against the electronic voting system. Applied computer forensics is that solution for conducting a computer forensics examination on an electronic voting system in real-time.

Using the ACF methodology, large amounts of data can quickly be analyzed speeding up the process of finding the proverbial needle-in-a-haystack, thereby making it both a proactive, reactive, and real-time process.

The combination of process automation and full forensic auditing makes up the ACF methodology and offers the most flexibility for a jurisdiction that is looking to maximize its resources, provide real-time monitoring and shorten the time frame for providing assurances that the operational integrity of the systems have remained intact.

The pillars of the ACF methodology are:

- Process Integration
- Baselining
- Understanding File Behavior
- Range of Change Management

## Process Integration

In order for the ACF methodology to work, it has to be integrated into the targeted process. In this case, the target is the election management process. The overall goal of ACF is to mitigate any threats to the normal operations of the electronic voting system that could negatively impact the outcome of an election. An understanding of the overall election management process as well as understanding the electronic voting

system's operations is necessary to identify aspects of the process where vulnerability could be exploited. The following illustration presents a high level threat model for a general election management process that could be applied to any election jurisdiction using an electronic voting system.

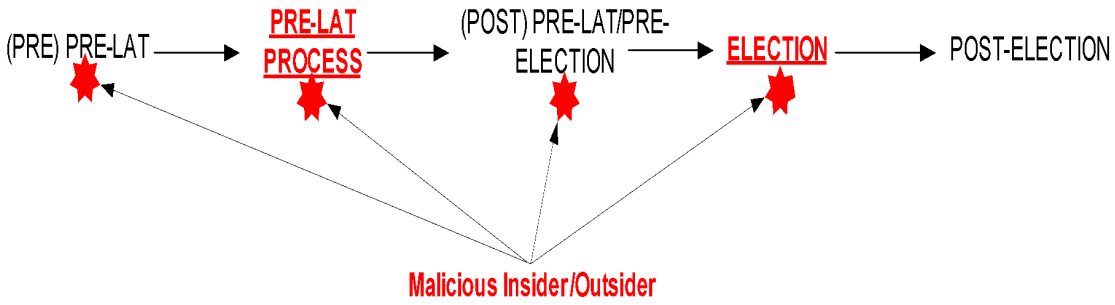


**Figure 2. High Level Threat Model for a General Election Management Process Illustration.**

Here, we have identified four prime areas in the election management process where a potential threat could be executed. In the (Pre) Pre-LAT (Logic and Accuracy Test) phase of the process, the majority of the system is generally at rest. This means that any direct voter-accessed equipment (such as DREs) is in its stored state. Components of the election management system that could be targeted are servers or other systems that would be initialized in preparation to start the Pre-LAT process. During this phase the ACF process could be implemented to ensure the operational integrity of the servers or any other system is intact. Implementing the ACF in this phase will mitigate any threat against the servers and stop the propagation of that threat to Pre-LAT phase. In this application of ACF, the current state of the server is compared against a pre-developed baseline for that particular server.

At this point in the process, an integrity initialization test can be conducted to verify that the integrity of all Pre-LAT supporting systems have been maintained while the systems were in a resting state between elections. This test will include comparing the previous Post-Election/Resting State phase of the server to the pre-developed baseline for the server to verify the following information:

- Static files were unchanged during the server's (Pre) Pre-LAT process unless they were changed by an authorized process and the execution of that process could be verified.
- If a dynamic file changed during the server's (Pre) Pre-LAT process, what were those changes and were they valid according the file's pre-defined range of change.
- If new files were added to the system. If yes, identify the files and identify if the file is malicious by examining the files' content.

During the Pre-LAT process, the majority of the electronic voting system is activated to begin overall preparations for the election. This means that DREs and other devices are prepared for field service. The

potential threat to the electronic voting system is magnified here because of the amount of people that the system is exposed to. The ACF process can be applied here to ensure that the integrity of the DREs and other field devices have been maintained during the Pre-LAT process.

At this point in the process an integrity verification/initialization test can be conducted to verify that the integrity of all the field service devices have been maintained during the Pre-LAT process. This test will include comparing the (Post) Pre-LAT state of the field service devices to the Baselines of those field devices to verify the following information:

- Static files were unchanged during the Pre-LAT process unless they were changed by an authorized process and the execution of that process could be verified.
- If a dynamic file changed during the Pre-LAT process, what were those changes and were they valid according the file's pre-defined range of change.
- If new files were added to the system. If yes, identify the files and identify if the file is malicious by examining the files' content.

In the (Post) Pre-LAT/Pre-Election phase of the process, the electronic voting system is at rest and awaiting the appointed time of the election to take place. At this point in the process an integrity verification test can be conducted to verify that the integrity of all Pre-LAT supporting systems has been maintained during the Pre-LAT process. This test will include comparing the (Pre) Pre-LAT state of the servers and other devices that were used to support Pre-LAT to the (Post) Pre-LAT/Pre-Election state of those same systems to verify the following information:

- Static files were unchanged during the Pre-LAT process unless they were changed by an authorized process and the execution of that process could be verified.
- If a dynamic file changed during the Pre-LAT process, what were those changes and were they valid according the file's pre-defined range of change.
- If new files were added to the system. If yes, identify the files and identify if the file is malicious by examining the files' content.

During the Election phase of the process, the entire electronic voting system is active. During this phase, an attack on the system could possibly come against any component or function of the system due to the fact that the system experiences its highest level of exposure. The ACF process can be applied to the Post-Election phase of the process.

At this point in the process an integrity verification test can be conducted to verify that the integrity of all electronic voting system components have been maintained during the election process. This test will include

a number of comparisons based on comparing the (Post) Pre-LAT state of the servers and all field service devices to Post-Election state of those same systems to verify the following information:

- Static files were unchanged during the Election process unless they were changed by an authorized process and the execution of that process could be verified.
- If a dynamic file changed during the Election process, what were those changes and were they valid according the file's pre-defined range of change.
- If new files were added to the system. If yes, identify the files and identify if the file is malicious by examining the files' content.

## Baselining

A key part of the ACF methodology is baselining. In order to manage the accuracy of the election's forensic analysis process, baselines of the electronic voting systems "clean state" must be captured. The "clean state" of the system the state in which every device is believed to have a clean, uncompromised software build installed on it. Baselining will be used to initially assess the integrity of the voting system before it goes into operation. This will provide assurances that the system is in a clean state before it is used in an election.

## Understanding File Behavior

Applications and operating systems are created with certain and finite functionality. As a result, every file that is part of the application or operating system (OS) has a specific purpose and function which determines what its 'file behavior' will be. Every file that is part of the application or OS has an expected behavioral footprint, meaning that the file is expected to behave in a certain way or as it was designed. Static files will generally behave in a static way. While not every function or capability of that static file is executed during routine system operations, the static file itself will not change at any time during routine system operations unless some other program function legitimately caused it to change such as program or system updates. Therefore, the behavior of static file is limited and can easily be determined. Dynamic files will behave in a dynamic way meaning that the contents of the dynamic file are designed to change based on the routine system operation that caused it to change. The presence of dynamic files should not be intimidating as generally, the range of change of the dynamic file is limited and based on the routine system operation, which is limited and as such, the range of change can be defined and measured. Because file behavior is limited based on the limited set of routine system operations, file behavior can be measured and captured and used

to validate future file behavior measurements to determine if those measurements are based on legitimate or malicious system activities.

## Range of Change Management

The concept of the 'range of change' is directly related to file behavior. The range of change is the specific delineation of the behavior of a file. Even though dynamic files are expected to change during normal routine system operations, it is expected that the range of change that a dynamic file would produce is finite and limited and can be defined. Therefore, if the changes to a dynamic file can be defined, verified and validated as legitimate, then illegitimate or malicious changes that are made to dynamic files can be easily identified enabling a proactive response and eradication before they could have a negative impact.

## A Practical Application of Applied Computer Forensics

Applied Computer Forensics (ACF) has been implemented since 2007 in one election jurisdiction as part of its overall election integrity management plan. This county's approach to implementing the ACF process focused on the operation of its core voting system in the following areas:

- Pre-LAT of the County's electronic voting system conducted at its warehouse.
- Central Tally servers located at it central office.
- Distributed Client Management systems located at it regional centers

The ACF process was integrated in a seamless manner with the least amount of disruption to the jurisdiction's election operations. As with any scientific approach, a control set of equipment that reflected basic system readiness as it was received by the election office from the manufacturer was used. By establishing a baseline from the control, all file changes and system activities were monitored as the voting system was prepared for use in the election. A combination of forensic analysis tools was used to identify and monitor the integrity of every file in the voting system. Based on the volume of data, automation was incorporated into each phase of the process in order to reduce costs and increase the detection speed of any system anomalies. Ten terabytes of information were collected from this jurisdiction's voting system and over six million files were analyzed.

At the completion of the 2008 Presidential Election Cycle, a full forensic audit report was presented to the jurisdiction verifying that no evidence of malicious code or tampering was found in its voting system. Although no evidence was found, this jurisdiction has taken the proactive step to manage any type of disaster that arises from the technical operation of their elections. Since 2006, this jurisdiction has positioned itself well with their ability to provide positive assurances that its election infrastructure operates with the utmost integrity.

Should something happen in the future, this jurisdiction will be able to readily respond by documenting all of the activities that transpired and apply lessons learned as part its process improvement plan.

## Conclusion

Voting systems are complex deployments with known vulnerabilities that are technology dependent. Process and procedures may open additional ones, or close some of these vulnerabilities. The EAC and state and local election jurisdictions are utilizing a number of tools to overcome these vulnerabilities and prevent and/or detect tampering with voting systems. Among these tools are hashing techniques, trusted build analysis and forensic analysis.

Forensic analysis is superior to both hash code testing and trusted build analysis because forensics is effective in finding anomalies both pre and post use of the software. With forensic analysis, software can be validated pre-use using proven algorithms. Dynamic files can be validated over time with post-use forensic analysis and forensic analysis can identify “foreign” code hidden in a system’s file structure. Forensic analysis is the only analytic method for software that can perform all of these functions.

As a result, if election jurisdictions incorporate forensics into their voting system monitoring programs, they would ensure efficient, robust and reliable vulnerability management of their election systems.

## Acknowledgements

### Edwin B. Smith, III

Cyrus Walker would like to acknowledge that significant portions of the introductory sections of this paper were authored by election industry colleague Edwin B. Smith, III of Longmont, CO. Mr. Smith's bio can be found on LinkedIn at http://www.linkedin.com/pub/ed-smith/1/a79/785. Mr. Walker would like to express his thanks to Mr. Smith for his expertise and invaluable contributions to this paper.

### Michelle M. Shafer

Cyrus Walker would like to thank Michelle M. Shafer of Austin, TX, Marketing Communications Consultant to Data Defenders, for her outstanding work in editing, producing and promoting this paper. Ms. Shafer's bio can be found on LinkedIn at http://www.linkedin.com/in/michelleshafer.