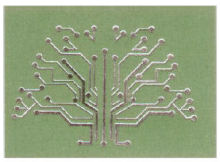

FREEMAN, CRAFT, MCGREGOR GROUP

Source Code Review

ES&S Voting System v 5.2.1.0

Report Date: 2017-08-28

Version: 1.1

Status: Released

Classification: Public

atsec information security corporation
9130 Jollyville Road, Suite 260
Austin, TX 78759
Tel: +1 512 615 7300
Fax: +1 512 615 7301
www.atsec.com

Revision history

Version	Change date	Author(s)	Changes to previous version
1.1	2017-08-28	Ryan Hill	Final

Trademarks

Table of Contents

List of tables

1 Executive Summary

This report was prepared by atsec information security corporation to review aspects of the security and integrity of the ES&S Voting System v. 5.2.1.0. atsec is an independent, third-party company providing information-security assurance related services.

This report identifies security weaknesses and vulnerabilities found through static code review and by searches of public vulnerability sources. The search focused particularly on those that could be exploited to alter vote recording, vote results, critical election data such as audit logs, or to conduct a denial of service attack on the voting system.

It should be noted that the public vulnerability search is most likely to identify vulnerabilities that have been reported in commonly used commercial off the shelf system components.

The static code analysis revealed 19 issues, the public vulnerability search identified 35 vulnerabilities that could potentially be used for an attack on the voting system. Of the 19 issues found by static code analysis, 3 were assessed to be of high severity, 10 were assessed to be of medium severity.

At a high level, weaknesses and vulnerabilities were identified that can be attributed to difficulties resulting from an aging and repeatedly maintained system. These include

- Failure to keep documentation up to date.
- Failure to upgrade cryptography as cryptographic algorithms become more susceptible to attacks over time. For example, the use of SHA-1 and other hashing algorithms, some key lengths and some random number generators are now widely recognized as being deprecated as is the use of CRC-32 for integrity checking of files.
- Inclusion of old code that may contain backdoors, and that is no longer used.
- Inconsistent design such as variable standards for password strength, the use of hard-coded passwords, and using common initialization vectors (IV)s for cryptographic functions.

In addition, numerous less severe but still noteworthy vulnerabilities were found related to code quality and non-conformance to the 2005 Voluntary Voting System Guidelines. See section 4 for all findings.

2 Introduction

This report was prepared by atsec information security corporation to review aspects of the security and integrity of the ES&S Voting System v 5.2.1.0. It has been prepared in support of a contract awarded to Freeman, Craft, McGregor Group, Inc. This project has a goal to provide voting system test support services to assist the California Secretary of State (SOS) with the evaluation of the ES&S Voting System v 5.2.1.0 (EAC Certification Number: ESSEVS5210) Voting System for its suitability for use in the State of California in accordance with Elections Code sections 19001 et seq.

The source code review was performed by the following atsec information security corporation consultants.

- Fiona Pattinson (Project Manager)
- King Ables (Lead Reviewer)
- Jason Gorgeoulis (Reviewer)
- Quentin Gouchet (Reviewer)
- Brandon Harvey (Reviewer)
- David Rumley (Reviewer)
- Swapneela Unkule (Reviewer)
- Brian Zhang (Reviewer)
- Ryan Hill (Documentation Specialist)

This document identifies the security vulnerabilities found through static code review and by searches of public vulnerability sources that could be exploited to alter vote recording, vote results, critical election data, such as audit logs, or to conduct a denial of service attack on the voting system.

2.1 Scope and Basis

The ES&S Voting System (EVS) v 5.2.1.0 (hereafter referred to as the “voting system” or simply as the “system”) is a paper-based voting system made up of the Election Management System (EMS), precinct tabulators, a Help America Vote Act compliant ballot marking device, and central count tabulators.

The system has the following software components:

- Election Management System (EMS), comprised of
  - Electionware,
  - Election Reporting Manager (ERM),
  - ExpressVote Previewer,
  - Voter Assist Terminal (VAT) Preview,
  - Event Log Service (ELS), and
  - Removable Media Service (RMS);
- ExpressVote and ExpressLink;
- DS200 Precinct Ballot Tabulator software;
- DS850 Central Ballot Tabulator software; and

- AutoMARK ADA Terminal Ballot Marking Device software.

The system can be setup to support one or more of the following hardware components:

- DS200 Precinct Ballot Tabulator,
- DS850 Central Ballot Tabulator,
- ExpressVote Universal Voting System, and
- AutoMARK Voting Assist Terminal.

atsec performed the source code review on the basis of an Agreement between Freeman, Craft, McGregor Group Inc., with the State of California, which states that the source code review includes examining the system in a manner that will provide the California Secretary of State with a basis for evaluating the extent to which the source code meets applicable standards. The threat model included in the Agreement is reproduced below and defines the threat parameters for the scope of this examination.

2.2 Inputs

The reviewers were provided with a Technical Data Package (TDP) including the source code and a set of documents that support the findings in this report. These documents were examined during the source code review to better understand the voting system and identify discrepancies between the documentation and the source code. These documents are listed in the References section.

2.3 Threat Model

This assessment is centered on the threat model given in the Request for Quotation (RFQ). The system is expected to counter the following attacks.

- Alter vote recording
- Alter vote results
- Alter critical election data, such as audit logs
- Conduct a denial of service attack on the voting system

To the extent possible, vulnerabilities found have been reported with an indication of whether the exploitation of the vulnerability would require access by any of the following.

- **Voter:** Usually has low knowledge of the voting machine design and configuration. Some may have more advanced knowledge. May carry out attacks designed by others. They have access to the machine(s) for less than an hour.
- **Poll worker:** Usually has low knowledge of the voting machine design and configuration. Some may have more advanced knowledge. May carry out attacks designed by others. They have access to the machine(s) for up to one week, but all physical security has been put into place before the machines are received.
- **Elections official insider:** Wide range of knowledge of the voting machine design and configuration. May have unrestricted access to the machine for long periods of time. Their designated activities include:
  - Set up and pre-election procedures
  - Election operation
  - Post-election processing of results
  - Archiving and storage operations

- **Vendor insider:** Has great knowledge of the voting machine design and configuration. They have unlimited access to the machine before it is delivered to the purchaser and, thereafter, may have unrestricted access when performing warranty and maintenance service, and when providing election administration services.

The atsec team did not attempt to demonstrate exploitability of identified potential vulnerabilities. However, identified potential vulnerabilities were described along with the anticipated factors necessary to mount an attack.

2.4 Methodology

The atsec team was tasked with the Source Code review which included, but was not limited to the following aspects.

- Evaluation of potential vulnerabilities and related issues (code quality and standards compliance), considering that an exploitable issue in a component that is not in itself security relevant could be used to subvert more critical data. This is an issue whenever the architecture of the system does not provide strong separation of the components.
- Adherence to the applicable standards in sections: 5 of Volume I, 7 of Volume I, and 5 of Volume II of the 2005 Voluntary Voting System Guidelines.
- Adherence to other applicable coding format conventions and standards including best practices for the coding language used, and any IEEE, NIST, ISO or NSA standards or guidelines which the Contractor find reasonably applicable.
- Analysis of the program logic and branching structure.
- Search for exposures to commonly exploited vulnerabilities, such as buffer overflows, integer overflow, inappropriate casting or arithmetic.
- Evaluation of the use and correct implementation of cryptography and key management.
- Analysis of error and exception handling.
- Evaluation of the likelihood of security failures being detected.
    - Are audit mechanisms reliable and tamper resistant?
    - Is data that might be subject to tampering properly validated and authenticated?
- Evaluation of the risk that a user can escalate his or her capabilities beyond those authorized.
- Evaluation of whether the design and implementation follow sound, generally accepted engineering practices. Is code defensively written to protect against:
    - Bad data;
    - Errors in other modules;
    - Changes in environment;
    - User errors; and
    - Other adverse conditions.
- Evaluation of whether the system is designed in a way that allows meaningful analysis, including:
    - Is the architecture and code amenable to an external review (such as this one)?
    - Could code analysis tools be usefully applied?

- Is the code complexity at a level that it obfuscates its logic?

- Search for embedded, exploitable code (such as “Easter eggs”) that can be triggered to affect the system.
- Search for dynamic memory access features which would permit the replacement of certificated executable code or control data or insertion of exploitable code or data.
- Search for use of runtime scripts, instructions, or other control data that can affect the operation of security relevant functions or the integrity of the data.

2.4.1 Potential vulnerabilities

The reviewers searched the following public list to identify vulnerabilities that may affect the system.

- MITRE Common Vulnerability and Exposures (CVEs) list https://cve.mitre.org/cve/cve.html

Although this list may not have entries for the voting system itself, constituent software and commercial off-the-shelf (COTS) components that the voting system integrates may contain vulnerabilities. The review team identified such components that the system relies upon and conducted searches for these products as well.

2.4.2 Code quality

While performing the examination of the code for other activities, the reviewers identified and recorded areas within the code base that demonstrate poor code quality. Although poor code quality does not necessarily identify vulnerabilities, it does provide an indication that vulnerabilities may exist.

The following coding standard was used during this analysis.

- 2005 Voluntary Voting System Guidelines [VVSG1], [VVSG2] and supplemental interpretation statements found at: http://www.eac.gov/testing_and_certification/request_for_interpretations1.aspx.

The reviewers also compared the code against software engineering best practices. Examples of best practices can be found in the following books:

- The CERT Oracle Secure Coding Standard for Java [CERTJ]
- The CERT C Security Coding Standard [CERTC].

These standards are based on accepted industry best practices in developing C/C++ code and in managed code (e.g., Java, J#, C#). There is no widely-accepted coding standard specific to COBOL.

The team also performed numerous informal static analysis activities on the source code to gather code quality data using customized command scripts.

2.4.3 Design

The source code review team used the technical data package, source code, and any material provided or otherwise publicly available to construct an understanding of the architecture and design of the voting system. This understanding included discovering the external interfaces and their security mechanisms and controls, particularly as much information as possible was gathered to support conclusions regarding the ability for a threat agent to tamper with or circumvent security controls.

Interfaces represent the primary attack surface of the voting system. Interfaces can include web-based interfaces, native graphic user interfaces, command line interfaces, or technical

interfaces that are not designed for direct user interaction (e.g., database connections). Each of these interfaces was examined to identify the security controls that counter the threats.

Secure interfaces also depend on filtering out poorly structured or corrupt data. The review team specifically checked for input validation mechanisms and determined if related attacks, such as command injection are possible.

2.4.4 Cryptography

While cryptography is often the most difficult security mechanism to break directly, misuse of cryptographic primitives can render that protection weak or non-existent. The review team identified where cryptography is used throughout the source code and determined if its use is appropriate for the given purpose. For example, using a cryptographic hash function to protect passwords is appropriate while using an encryption algorithm with a hard-coded key is not.

2.4.5 Back doors

Those with access to the voting system during development and having malicious intent can place back doors into the source code so that they could gain unauthorized access to the voting system during operation. Back doors are extremely hard to find because a seasoned programmer can obfuscate code to look benign.

The review team marked areas of vulnerabilities as identified above for further scrutiny. For example, a particular area of code with poor code quality and access to sensitive information such as authentication credentials might be a good place to hide a back door. The reviewers gave such areas with extra scrutiny by considering insider threats in addition to unintentional implementation flaws.

2.4.6 Measurement of findings

A summary of findings is listed in section 4. Each finding contains the following information.

- A description of the vulnerability or weakness
- An assessment of what threats are involved in the possible exploitation of the vulnerability or weakness
- A categorization of the findings, which can be:
    - A weakness in the source code. Weaknesses are issues identified in the source code that are not directly exploitable but may indicate the existence of exploitable vulnerabilities within the source code.
    - A non-conformity in the code quality standards. Non-conformities do not necessarily imply weaknesses, though the rationale for the requirement is often based on preventing weaknesses.
    - A potential vulnerability in the source code. The reviewers consider potential vulnerabilities to likely be exploitable.
    - A vulnerability in the source code. The reviewers have either shown or have referenced other parties who have asserted the vulnerability to be exploitable.
- A severity level of the findings, which can be either:
    - A low severity finding. Low severity implies either the impact to the product is low or already mitigated by the system, or the difficulty in exploitation would likely require unrestricted access to the systems, expert knowledge of the system, or would require cost prohibitive resources.
    - A medium severity finding. Medium severity implies either the impact of exploitation to the product would be significant, or the difficulty in exploitation

would likely require extended access to the systems, informed knowledge of the system, or would require significant resources.

- A high severity finding. High severity implies either the impact of exploitation to the product would result in complete compromise of security, or the difficulty in exploitation would likely require little to no access or knowledge of the systems or little to no resources.

2.4.7 Depth of analysis

Because of the complexity and volume of the material to be reviewed, limited time available and broad scope (assessment of documents and quality of the code, along with source code review), the team concentrated on surveying a breadth of categories of vulnerabilities that they could identify, and only reviewed in depth enough samples of each of the categories to determine how that vulnerability was being handled. For all the categories, no attempt was made to enumerate how many instances existed. Other source code review projects would be likely to find more instances, but those findings should be within the listed categories.

3 Description of the ES&S Voting System

The ES&S Voting System is a suite of software and hardware components for conducting and reporting elections.

3.1 Voting System Functions

The ES&S Voting System provides a number of high-level functions necessary to conduct an election. These activities include the following.

- Creation and definition of ballots
- Programming of precinct scanners and other hardware
- Tabulation and reporting of election results
- Audit logs generated for operations, ballots, and user activities

Precinct scanners are used for processing ballots at each polling place. Election data is gathered for tabulation and reporting, and an audit record is generated for all activities. Election data is transferred between components via physical move of compact flash and USB flash drives.

3.2 Physical Components

Several components are used in conducting an election with ES&S EVS. Some are specialized hardware components built or assembled by ES&S, others are COTS products used to run ES&S EVS. The following are the specialized hardware components.

DS200 Precinct Ballot Tabulator—a touchscreen digital scanner that is designed to scan each paper ballot at the polling site. The image-scanner uses a set of two high-resolution cameras to simultaneously image the front and back of a ballot. The resulting ballot images are then decoded by a recognition engine. The tabulator stores current totals in both internal memory and on removable USB flash memory. It can also produce reports from the scanner's internal printer.

DS850 Central Ballot Tabulator—a high-speed digital scanner and ballot counter. While scanning, the DS850 will print a continuous audit log to an attached printer. The scanner stores all of the scanned data internally and to results collection media that officials can use to format and print results. The results collection media can be transferred to a PC that is running the Election Reporting Manager (ERM) software.

ExpressVote Universal Voting System—a universal vote capture device usable by all voters, including those with special needs. It combines paper-based voting and touch screen technology and produces an independent, voter-verifiable paper record that is digitally scanned for tabulation by ES&S ballot scanners.

AutoMARK Voting Assist Terminal—an optical scan ballot marker that assists voters with disabilities.

3.3 Logical Components

The Election Management System (EMS) is a set of applications used during the pre-election design phase and post-election tabulation phase of an election.

3.3.1 Election Management System software

EMS client and server components can run on the same physical server (including the back-end server) or on one or more separate platforms. EMS components operate on a stand-alone, hardened system.

Access control to EMS client components is role-based where each user is a member of a single pre-defined role, created and managed by an EMS election administrator. Each role has its own set of permissions that govern what operations the user is allowed to perform.

Electionware provides election management functions allowing jurisdictions to manage an entire election from creating ballots to reporting results.

Election Reporting Manager (ERM) is used to consolidate ballot tabulations and report totals. It is able to print reports of polling and cast ballot results.

ExpressVote Previewer provides ability to preview audio and screen layout for ExpressVote prior to creation of media files.

VAT Preview lets the user preview text, audio and the layout of the screen before downloading that data to the AutoMARK.

Event Log Service (ELS) is a Windows service supporting all EVS applications running on the server.

Removable Media Service (RMS) supports installation and removal of election data on removable media.

3.3.2 Supporting software

The voting system is designed to use several COTS software components.

- Microsoft Windows 7 Professional 64-bit SP 1 for standalone and client workstations
- Microsoft Windows Server 2008 R2 64-bit SP 1 for EMS and results servers
- Microsoft Windows patches (via WSUS 8.8 offline utility)
- Adobe Acrobat XI
- RM/Cobol Runtime 12.06
- Symantec Endpoint Protection Small Biz Ed 2013 12.1.4 64-bit
- Symantec Endpoint Protection Intelligent Updater 6.5 Premium

3.4 Interfaces

The voting system moves data between external interfaces and internal components in a variety of ways: peripheral devices, files, and databases. This section will discuss these interfaces, the types of data, and where the data goes.

3.4.1 Network interfaces

The Electionware workstation has a local Ethernet connection to function in a protected data center environment.

The only tabulation device with an Ethernet connection is the DS850.

The DS200 includes an RJ-11 modem connection.

3.4.2 Peripheral devices

Data is transferred between the systems using the appropriate media. The appropriate media varies depending on the device that is receiving the data. The appropriate media depends on the tabulator that the data is being transferred to/from. Table 1 summarizes the appropriate media for each device.

Device	Appropriate Media
DS200	Compact Flash Card, USB Memory Stick
DS850	USB Memory Stick
ExpressVote	USB Memory Stick
AutoMark	Compact Flash Card

Table 1: EVS Device Media

Ballot layout data will be transferred from Electionware to the appropriate tabulators. Election results data will be transferred from the appropriate tabulators to the ERM.

The ERM can be configured to have a monitor display the ballot data as it is being tabulated by the DS200 and DS850.

4 Findings

4.1 Public Vulnerability Search

Table 2 lists the CVEs identified that could potentially impact the voting system.

CVE	Description	Rationale
CVE-2012-4575	The add_database function in objects.c in the pgbouncer pooler 1.5.2 for PostgreSQL allows remote attackers to cause a denial of service (daemon outage) via a long database name in a request.	The system uses PostgreSQL. Hence the CVE may be applicable.
CVE-2005-3656	Multiple format string vulnerabilities in logging functions in mod_auth_pgsql before 2.0.3, when used for user authentication against a PostgreSQL database, allows remote unauthenticated attackers to execute arbitrary code, as demonstrated via the username.	The system uses PostgreSQL. Hence the CVE may be applicable.
CVE-2016-5424	PostgreSQL before 9.1.23, 9.2.x before 9.2.18, 9.3.x before 9.3.14, 9.4.x before 9.4.9, and 9.5.x before 9.5.4 might allow remote authenticated users with the CREATEDB or CREATEROLE role to gain superuser privileges via a (1) " (double quote), (2) \ (backslash), (3) carriage return, or (4) newline character in a (a) database or (b) role name that is mishandled during an administrative operation.	The system uses PostgreSQL 9.1.9-3. Hence the CVE may be applicable.
CVE-2017-0014	The Windows Graphics Component in Microsoft Office 2010 SP2; Windows Server 2008 R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via a crafted web site, aka "Windows Graphics Component Remote Code Execution Vulnerability." This vulnerability is different from that described in CVE-2017-0108.	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-7246	The kernel-mode drivers in Microsoft Windows Server 2008 R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, Windows 10 Gold, 1511, and 1607, and Windows Server 2016 allow local users to gain privileges via a crafted application, aka "Win32k Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-7238	Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, Windows 10 Gold, 1511, and 1607, and Windows Server 2016 mishandle caching for NTLM password-change requests, which allows local users to gain privileges via a crafted application, aka "Windows NTLM Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.

CVE	Description	Rationale
CVE-2016-7205	Animation Manager in Microsoft Windows Server 2008 R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, Windows 10 Gold, 1511, and 1607, and Windows Server 2016 allows remote attackers to execute arbitrary code via a crafted web site, aka "Windows Animation Manager Memory Corruption Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-3230	The Search component in Microsoft Windows 7, Windows Server 2008 R2 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold and 1511 allows local users to cause a denial of service (performance degradation) via a crafted application, aka "Windows Search Component Denial of Service Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-3226	Active Directory in Microsoft Windows Server 2008 R2 SP1 and Server 2012 Gold and R2 allows remote authenticated users to cause a denial of service (service hang) by creating many machine accounts, aka "Active Directory Denial of Service Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-0176	dxgkrnl.sys in the DirectX Graphics kernel subsystem in the kernel-mode drivers in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold and 1511 allows local users to gain privileges via a crafted application, aka "Microsoft DirectX Graphics Kernel Subsystem Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-0101	Microsoft Windows Server 2008 R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold and 1511 allow remote attackers to execute arbitrary code via crafted media content, aka "Windows Media Parsing Remote Code Execution Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-0098	Microsoft Windows Server 2008 R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 allow remote attackers to execute arbitrary code via crafted media content, aka "Windows Media Parsing Remote Code Execution Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-6130	Integer underflow in Uniscribe in Microsoft Windows 7 SP1 and Windows Server 2008 R2 SP1 allows remote attackers to execute arbitrary code via a crafted font, aka "Windows Integer Underflow Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.

CVE	Description	Rationale
CVE-2015-2473	Untrusted search path vulnerability in the client in Remote Desktop Protocol (RDP) through 8.1 in Microsoft Windows 7 SP1 and Windows Server 2008 R2 SP1 allows local users to gain privileges via a Trojan horse DLL in the current working directory, as demonstrated by a directory that contains a .rdp file, aka "Remote Desktop Protocol DLL Planting Remote Code Execution Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-2368	Untrusted search path vulnerability in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8.1, Windows Server 2012 R2, and Windows RT 8.1 allows local users to gain privileges via a Trojan horse DLL in the current working directory, aka "Windows DLL Remote Code Execution Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-2366	win32k.sys in the kernel-mode drivers in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allows local users to gain privileges via a crafted application, aka "Win32k Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-1635	HTTP.sys in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, and Windows Server 2012 Gold and R2 allows remote attackers to execute arbitrary code via crafted HTTP requests, aka "HTTP.sys Remote Code Execution Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-0098	Task Scheduler in Microsoft Windows 7 SP1 and Windows Server 2008 R2 SP1 allows local users to gain privileges by triggering application execution by an invalid task, aka "Task Scheduler Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-0084	The Task Scheduler in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 does not properly constrain impersonation levels, which allows local users to bypass intended restrictions on launching executable files via a crafted task, aka "Task Scheduler Security Feature Bypass Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.

CVE	Description	Rationale
CVE-2015-0062	Microsoft Windows Server 2008 R2 SP1, Windows 7 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allow local users to gain privileges via a crafted application that leverages incorrect impersonation handling in a process that uses the SeAssignPrimaryTokenPrivilege privilege, aka "Windows Create Process Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-0059	win32k.sys in the kernel-mode drivers in Microsoft Windows Server 2008 R2 SP1, Windows 7 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allows local users to gain privileges via a crafted TrueType font, aka "TrueType Font Parsing Remote Code Execution Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-0016	Directory traversal vulnerability in the TS WebProxy (aka TSWbPrxy) component in Microsoft Windows Vista SP2, Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allows remote attackers to gain privileges via a crafted pathname in an executable file, as demonstrated by a transition from Low Integrity to Medium Integrity, aka "Directory Traversal Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2015-0002	The AhcVerifyAdminContext function in ahcache.sys in the Application Compatibility component in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 does not verify that an impersonation token is associated with an administrative account, which allows local users to gain privileges by running AppCompatCache.exe with a crafted DLL file, aka MSRC ID 20544 or "Microsoft Application Compatibility Infrastructure Elevation of Privilege Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2014-2814	Microsoft Service Bus 1.1 on Microsoft Windows Server 2008 R2 SP1 and Server 2012 Gold and R2 allows remote authenticated users to cause a denial of service (AMQP messaging outage) via crafted AMQP messages, aka "Service Bus Denial of Service Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.

CVE	Description	Rationale
CVE-2014-0316	Memory leak in the Local RPC (LRPC) server implementation in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allows remote attackers to cause a denial of service (memory consumption) and bypass the ASLR protection mechanism via a crafted client that sends messages with an invalid data view, aka "LRPC ASLR Bypass Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2014-0263	The Direct2D implementation in Microsoft Windows 7 SP1, Windows Server 2008 R2 SP1, Windows 8, Windows 8.1, Windows Server 2012 Gold and R2, and Windows RT Gold and 8.1 allows remote attackers to execute arbitrary code via a large 2D geometric figure that is encountered with Internet Explorer, aka "Microsoft Graphics Component Memory Corruption Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2013-3902	Use-after-free vulnerability in win32k.sys in the kernel-mode drivers in Microsoft Windows Server 2008 R2 SP1 and Windows 7 SP1 on 64-bit platforms allows local users to gain privileges via a crafted application, aka "Win32k Use After Free Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2013-3881	win32k.sys in the kernel-mode drivers in Microsoft Windows 7 SP1 and Windows Server 2008 R2 SP1 allows local users to gain privileges via a crafted application, aka "Win32k NULL Page Vulnerability."	The system uses Windows Server 2008 R2 SP1. Hence the CVE may be applicable.
CVE-2016-3230	The Search component in Microsoft Windows 7, Windows Server 2008 R2 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold and 1511 allows local users to cause a denial of service (performance degradation) via a crafted application, aka "Windows Search Component Denial of Service Vulnerability."	The system uses Microsoft Windows 7. Hence the CVE may be applicable.
CVE-2013-3862	Double free vulnerability in Microsoft Windows 7 and Server 2008 R2 SP1 allows local users to gain privileges via a crafted service description that is not properly handled by services.exe in the Service Control Manager (SCM), aka "Service Control Manager Double Free Vulnerability."	The system uses Microsoft Windows 7. Hence the CVE may be applicable.
CVE-2013-3154	The signature-update functionality in Windows Defender on Microsoft Windows 7 and Windows Server 2008 R2 relies on an incorrect pathname, which allows local users to gain privileges via a Trojan horse application in the %SYSTEMDRIVE% top-level directory, aka "Microsoft Windows 7 Defender Improper Pathname Vulnerability."	The system uses Microsoft Windows 7. Hence the CVE may be applicable.

CVE	Description	Rationale
CVE-2013-2554	Unspecified vulnerability in Microsoft Windows 7 allows attackers to bypass the ASLR and DEP protection mechanisms via unknown vectors, as demonstrated against Firefox by VUPEN during a Pwn2Own competition at CanSecWest 2013, a different vulnerability than CVE-2013-0787.	The system uses Microsoft Windows 7. Hence the CVE may be applicable.
CVE-2013-2553	Unspecified vulnerability in the kernel in Microsoft Windows 7 allows local users to gain privileges via unknown vectors, as demonstrated by Nils and Jon of MWR Labs during a Pwn2Own competition at CanSecWest 2013, a different vulnerability than CVE-2013-0912.	The system uses Microsoft Windows 7. Hence the CVE may be applicable.
CVE-2014-9228	sysplant.sys in the Manager component in Symantec Endpoint Protection (SEP) before 12.1.6 allows local users to cause a denial of service (blocked system shutdown) by triggering an unspecified deadlock condition.	The system uses Symantec Endpoint Protection 12.1.4. Hence the CVE may be applicable.
CVE-2014-9227	Multiple untrusted search path vulnerabilities in the Manager component in Symantec Endpoint Protection (SEP) before 12.1.6 allow local users to gain privileges via a Trojan horse DLL in an unspecified directory.	The system uses Symantec Endpoint Protection 12.1.4. Hence the CVE may be applicable.

Table 2: CVEs identified during the public vulnerability search

4.2 Static Code Analysis & Documentation Review

Table 3 summarizes the findings that arose from the source code review team's assessment of the voting system. Potential exploitation of a weakness or vulnerability and type of attacker is noted where applicable.

ID	Description	Assessment	Categorization
001	Usage of SHA-1 for signature generation	The file Sign.cs makes the use of SHA-1 for signature generation. Some weaknesses have been recently found in the SHA-1 algorithm and it recommended to use a hash from the SHA-2 family (i.e. SHA-256 or above). Moreover, signature generation using SHA-1 is not approved by NIST.	**Type:** potential vulnerability and non-conformity (FIPS) **Severity:** medium
002	Too small key size for RSA key and signature generation	The file Sign.cs implements of RSA key generation, and signature generation operation which uses 1024-bit RSA keys. That value is hard coded in the code on line 67. 1024-bit RSA keys are not recommended to use. Moreover, NIST disallows 1024-bit RSA keys for key	**Type:** potential vulnerability and non-conformity (FIPS) **Severity:** medium

ID	Description	Assessment	Categorization
		generation and signature generation.	
003	Typos in code comments	Typos in code comments	**Type:** non-conformity **Severity:** low
004	Derive symmetric key from a simple hash	An AES-256 key is derived from the password by hashing the password with SHA-256. A proper KDF should be used instead, such as PBKDF2.	**Type:** potential vulnerability **Severity:** medium
005	Usage of the DES algorithm	The DES algorithm does not provide enough security strength. Could be replaced by Triple-DES, with 3 independent key, to make it compliant with NIST requirements, and avoid re-implementing another algorithm.	**Type:** potential vulnerability and non-conformity (FIPS) **Severity:** medium
006	Usage of RIPEMD-160	The RIPEMD algorithm provides only 80-bit of security strength, which is considered insufficient for today's computation capabilities. Should be replaced by a hash algorithm from the SHA-2/3 family.	**Type:** potential vulnerability and non-conformity (FIPS) **Severity:** medium
007	Incorrect seeding of a DRBG	The used RNG is a SP 800-90A DRBG according to the comment on line 39 (currently the only approved RNGs are the SP 800-90A DRBGs' implementations). However, on line 71, it looks the seed length is 40 bytes, so 320 bits. This is too small according to SP 800-90A, depending on which DRBG Type is used. CTR_DRBG with Triple-DES, AES-128 or AES-192 is compliant. CTR_DRBG with AES-256, Hash_DRBG or HMAC_DRBG is not compliant.	**Type:** potential vulnerability and non-conformity (FIPS) **Severity:** medium
008	Usage of /dev/urandom as a TRNG for seeding a DRBG	/dev/urandom provides the seed for the RNG for what seems to be a SP 800-90A DRBG implementation. /dev/urandom is an interface to obtain random data from the Linux Kernel RNG. As opposed to /dev/random, /dev/urandom might not hold enough entropy to seed the DRBG especially on a headless Linux system with not much interaction (in the case of the DS200, no networking, no keyboard or mouse). This could be changed to /dev/random, or another noise source could be added to the system. Moreover, an entropy analysis of the	**Type:** potential vulnerability **Severity:** medium

ID	Description	Assessment	Categorization
		Linux kernel RNG on a DS200 could be performed to assess the entropy of the random data inside the Linux kernel on this platform.	
009	Usage of MD2 and MD5	According to line 189 and 190, MD2 and MD5 can be used. These algorithms have been found to not provide enough security strength. A hash algorithm from the SHA-2/3 family should be used instead.	**Type:** vulnerability **Severity:** medium
010	AutoMARK EEPROM password present in the code	In case the macro 'BYPASS_PASSWORD' exists, a deterministic password will be used for EEPROM.	**Type:** weakness **Severity:** low

011	Code handles a bug but may not be fixing the source of the problem	There are isolated areas where the reviewers identified try-catch blocks that may not be robust to all potential errors.
		There are cases where try-catch is implemented with no arguments provided in the catch block. For example, in RSASecurityLibrary.cpp, a catch block is handled as follows.
		try
		{
		if (verstr!=(TCHAR *)NULL) { /* check argument */
		wcsncpy(verstr,RSA_LIB_VERSION,127); /* copy version string */
		// 127 is max allowed length to go into verstr
		}
		}
		catch(...)
		{
		}When this occurs, the catch block handles all potential errors with the same code. Using this type of catch block cannot find the exact type of error that has been thrown in the try block.
		An exception is a response to an exceptional circumstance that arises while a program is running. When a specific exception is caught, it means
			Type: weakness
			Severity: low

ID	Description	Assessment	Categorization
		that the developer more concretely understands the behavior of the application/code and has more control over it. Other than the exception thrown by the specific code in the try block, exception can occur in cases such as out of memory, stack overflow or even bug in the code implementation. In such scenarios, catching all exceptions can lead to a case where the code handles a bug but it's not fixing the source of the problem.	
012	Not meeting Default Case requirements in Section 5 Volume II of the 2005 Voluntary Voting System Standards	Requirement: Review of the code should ensure that for those languages supporting case statements, a default choice is explicitly defined to catch values not included in the case list. ExpressVote 2015-11-09\ExpressVote_1.4.1.0l_Source\ExpressVote\astro\astroData\C#\Hardware\Eeprom.cs switch block on line 1596	**Type:** non-conformity (VVSG) **Severity:** low
013	Use of fixed/default IV value for encryption/decryption	EMS 2015-11-02\electionware_4.7.1.0n_SourcePkg\Control\src\com\essvote\util\crypto\CryptoJUtil.java encrypt() and decrypt() function makes use of hardcoded IV for AES encryption/decryption In the same code, encryptBundle() function uses this encrypt() function to encrypt the data and then it is stored. Also the encrypt() function is used in EMS 2015-11-02\electionware_4.7.1.0n_SourcePkg\Results\src\com\essvote\ electionWare\results\result\dsloader\SecurityKeys.java by the function getPrivateKey() to retrieve the encrypted private key. Based on the code comments, this key is used to decrypt election qualification code. Using the same IV for all data is equivalent to not using an IV. The main consequence of reusing the IV is that if two messages begin with the same sequence of bytes then the encrypted messages will also be identical for a few blocks. This leaks data and opens the	**Type:** potential vulnerability **Severity:** medium

ID	Description	Assessment	Categorization
		possibility of some attacks. E.g. if an attacker can set their own credentials and observe the ciphertext that's generated, they can compare it with other encrypted credentials to find out information about the key.	
014	Source code vulnerabilities	We used the open source application Flawfinder to inspect the source code and search for potential vulnerabilities. The source code is available at https://www.dwheeler.com/flawfinder/. Page 9 of Flawfinder's documentation (https://www.dwheeler.com/flawfinder/flawfinder.pdf) lists the CWEs (COMMON WEAKNESS ENUMERATION) that Flawfinder covers in its search: • CWE-20: Improper Input Validation • CWE-22: Improper Limitation of a Pathname to a Restricted Directory (''Path Traversal'') • CWE-78: Improper Neutralization of Special Elements used in an OS Command (''OS Command Injection'')* • CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer (a parent of CWE-120*, so this is shown as CWE-119:CWE-120) • CWE-120: Buffer Copy without Checking Size of Input (''Classic Buffer Overflow'')* • CWE-126: Buffer Over-read • CWE-134: Uncontrolled Format String* • CWE-190: Integer Overflow or Wraparound* • CWE-250: Execution with Unnecessary Privileges • CWE-327: Use of a Broken or Risky Cryptographic Algorithm* • CWE-362: Concurrent Execution using Shared Resource with Improper Synchronization (''Race Condition'')	**Type:** potential vulnerability **Severity:** medium

ID	Description	Assessment	Categorization
		• CWE-377: Insecure Temporary File • CWE-676: Use of Potentially Dangerous Function* • CWE-732: Incorrect Permission Assignment for Critical Resource* • CWE-785: Use of Path Manipulation Function without Maximum-sized Buffer (child of CWE-120*, so this is shown as CWE-120/CWE-785) • CWE-807: Reliance on Untrusted Inputs in a Security Decision* • CWE-829: Inclusion of Functionality from Untrusted Control Sphere* The output of Flawfinder returned 2483 hits (i.e. Found CWEs). An example of each found CWE is given in the work paper.	
015	Audit records do not have sufficient safeguards to prevent alteration by a compromised or malicious administrator	The owner of the audit table is currently set to the default database administrator “ewsergtsop” in the auditSchema.sql asset. In the event of a malicious or compromised “ewsergstop” breach, audit records and/or audit functionality can be altered with no resulting evidence trail. The audit schema for all versions of Electionware EMS is at risk of failed integrity in the event of successful privilege escalation to the built-in administrator account or malicious insider. A separate and distinct owner of the audit table would establish effective principles of least privilege.	**Type:** potential vulnerability **Severity:** high
016	Use of SECURITY DEFINER in PLPGSQL stored procedures may allow privilege escalation	When SECURITY DEFINER Is invoked in Postgres functions the permissions for the owner of the function are granted regardless of the permissions of the user invoking the function. SECURITY DEFINER is used in stored procedure .sql files throughout the ElectionWare suite. This constitutes a very large attack surface should method to invoke these functions be found by an	**Type:** potential vulnerability **Severity:** high

ID	Description	Assessment	Categorization
		unauthorized party. 2427 instances of SECURITY DEFINER were found in the Electionware /DataSprocsPkg directories which include administrative function for credential and user management, assets and schema owners, and many UPDATE and INSERT capabilities. The use of this privilege management may need to be more tightly controlled by assigning segregated function owners or reduced in scope to mitigate the attack potential.	
017	Use of weak default password for database administrative user	The default password for the database administrator is exposed in plain text and follows common substitution methods, which drastically decreases the difficulty of cracking. If this finding is mitigated by forcing a change of this password then no action is required. If a change is not required and enforced by some other method then corrective measures are needed.	**Type:** potential vulnerability **Severity:** high
018	Hard coded password	There is a hard-coded password set for the grub boot loader. It might be changed later doing the install process, but this could present an issue. All hard-coded passwords in source code or scripts are of concern.	**Type:** weakness **Severity:** low
019	Leap year calculation	Leap year calculations in Java code does not include some exceptions. Errors will occur in the future.	**Type:** weakness **Severity:** low

Table 3: Summary of issues discovered during the static code analysis

Glossary

AES	Advanced Encryption Standard
API	Application Programming Interface
CAPI	Crypto API
CBC	Cipher Block Chaining
CMVP	Cryptographic Module Validation Program
COTS	Commercial Off-The-Shelf
CRC	Cyclic Redundancy Check
CTR	Counter
CVE	Common Vulnerability and Exposures
CVR	Cast Vote Record
CWE	Common Weakness Enumeration
TDES	Triple-Data Encryption Standard
EC	Elliptic Curve
ECDSA	Elliptic Curve Digital Signature Algorithm
EDM	Election Data Manager
ELS	Event Log Service
EMS	Election Management System
EQC	Election Qualification Code
ERM	Election Reporting Manager
EVS	ES&S Voting System
FIPS	Federal Information Processing Standard
HMAC	Hash Message Authentication Code
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IP	Internet Protocol
IV	Initialization Vector
KDF	Key Derivation Function
LAN	Local Area Network
OS	Operating System
PBKDF	Password-Based Key Derivation Function
PC	Personal Computer
PKI	Public Key Infrastructure
PRF	Pseudo-Random Function
PRNG	Pseudorandom Number Generator
RCV	Ranked Choice Voting

RMS	Removable Media Service
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standards
SOS	Secretary of State
TCP	Transmission Control Protocol
TDP	Technical Data Package
USB	Universal Serial Bus
VAT	Voter Assist Terminal
VVSG	Voluntary Voting System Guidelines
XML	Extensible Markup Language

References

Documentation provided for the source code review included ES&S EVS product documentation and other publically available standards documents. The atsec source code review team also consulted other publically available documents listed in the last group.

ES&S Documents

[AMSOG] AutoMARK System Operator's Guide, Document Version 1.0, Firmware Version 1.9, Published September 18, 2015

[AMSSP] AutoMARK System Security Specifications, Revision 7, 2015-09-08

[DS200OG] DS200 Operator's Guide, Document Version 1.1, Software Version 2.12, Published October 16, 2015

[DS200SDS] DS200 – Software Design Specification, ES&S Voting System, Revision 1.0, 2015

[DS850OG] DS850 Operator's Guide, Document Version 1.1, Software Version 2.10, Published October 12, 2015

[DS850SDS] DS850 – Software Design Specification, ES&S Voting System, Revision 1.0, 2015

[ELOG] ExpressLink, Operator's Guide, Firmware Version 1.3, Manual Revision 1.1, Released October 26, 2015

[ELSSDS] Election Systems & Software, ES&S Software Design Specifications, Event Log Service, Revision 1.0, 2015-09-18

[ELSUG] ES&S EVS Event Logging Service User's Guide, Document Version 1.0, Software Version 1.5, Published September 18, 2015

[ERMSDS] Election Systems & Software, ES&S Software Design Specifications, ERM 8.12.1.0, ES&S Voting System 5.2.1.0, Revision 1.0, 2015-09-18

[ERMSDSA] Election Systems & Software, ES&S Software Design Specifications, ERM 8.12.1.0 Appendices, ES&S Voting System 5.2.1.0, Revision 1.0, 2015-09-18

[ERMUG] Election Reporting Manager User's Guide, Document Version 1.1, Software Version 8.12, Published October 9, 2015

[ESSCS] ES&S Standards and Procedures, Coding Standards, Revision 3.0, 2014-04-05

[ESSSP] ES&S Standards and Procedures, ES&S System Development Program, Revision 2.0, 2014-04-05

[ESSVSSS] ES&S Voting System 5.2.1.0, Voting System Security Specification, Revision 1.0, 2015-09-25

[EVOG] ExpressVote, Operator's Guide, Firmware Version 1.4, Revision 1.2, Released October 28, 2015

[EVOGA] ExpressVote, Operator's Guide Appendices, Firmware Version 1.4, Revision 1.0, Released September 18, 2015

[EVPSDS] EVS5210, ExpressVote Software Design and Specification, Revision 1.0, 2015-09-18

[EWSDS] EVS5210, Electionware – Software Design and Specification, Revision 1.0, 2015-09-18

[EWV1EG] Electionware Volume I: Administrators Guide, Document Version 1.1, Software Version 4.7, Published October 7, 2015

[EWV2DUG] Electionware Volume II: Define User's Guide, Document Version 1.1, Software Version 4.7, Published October 6, 2015

[EWV3DUG] Electionware Volume III: Design User's Guide, Document Version 1.1, Software Version 4.7, Published October 7, 2015

[EWV4DUG] Electionware Volume IV: Deliver User's Guide, Document Version 1.2, Software Version 4.7, Published October 29, 2015

[EWV5RUG] Electionware Volume V: Results User's Guide, Document Version 1.0, Software Version 4.7, Published September 18, 2015

Public Documents

[CERTC] Seacord, Robert C., The CERT C Secure Coding Standard, Addison-Wesley, Upper Saddle River, NJ, 2009

[CERTJ] Long, et al., The CERT Oracle Secure Coding Standard for Java, Addison-Wesley, Upper Saddle River, NJ, 2012

[FIPS140IG] National Institute of Standards and Technology, Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program, January 2016, http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf

[FIPS140-2] National Institute of Standards and Technology, FIPS 140-2 Security Requirements for Cryptographic Modules, May 2001, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[FIPS140-2A] National Institute of Standards and Technology, FIPS 140-2 Annex A: Approved Security Functions, December 2002, http://csrc.nist.gov/publications/fips/fips140-2/fips1402annexa.pdf

[FIPS180-4] National Institute of Standards and Technology, FIPS 180-4 Secure Hash Standard (SHS), March 2012, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[FIPS186-4] National Institute of Standards and Technology, FIPS 186-4 Digital Signature Standard (DSS), July 2013, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[FIPS197] National Institute of Standards and Technology, FIPS 197 Advanced Encryption Standard, November 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[FIPS198-1] National Institute of Standards and Technology, FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC), July 2008, http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[NIST57] National Institute of Standards and Technology, NIST Special Publication 800-57, Recommendation for Key Management—Part 1: General (Revised), January 2016, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf

[NIST90A] National Institute of Standards and Technology, NIST Special Publication 800-90A, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June, 2015, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf

[NIST131A] National Institute of Standards and Technology, NIST Special Publication 800-131A, Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths, January, 2011, http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

[VVSG1] United States Election Assistance Commission, 2005 Voluntary Voter System Guidelines, Volume 1, Version 1.0, 2005 http://www.eac.gov/testing_and_certification/voluntary_voting_system_guidelines.aspx

[VVSG2] United States Election Assistance Commission, 2005 Voluntary Voter System Guidelines, Volume 2, Version 1.0, 2005 http://www.eac.gov/testing_and_certification/voluntary_voting_system_guidelines.aspx