SysTest Labs
216 16th Street, Suite 700
Denver, CO 80202
p: 303.575.6881 | f: 303.575.688
www.systest.com

SysTest labs®

November 17, 2009

Review of Technical Data Packages (TDPs)

FOR

State of New York
State Board of Elections

Document Revision History

The following is a record of the changes that have occurred in this document since the time of its original submission.

Version	*Change Description*	*Author(s)*	*Date*
1.0	• Internal Review	James “Jet” Henry	16-Nov-09
1.1	• Review of Technical Data Packages submitted to NYSBOE for Acceptance (Deliverable 5)	Mark Phillips James “Jet” Henry	17-Nov-09

Table of Contents

List of Tables

1 PROJECT DESCRIPTION

1.1 Results Summary

This Technical Data Package (TDP) review was conducted for the New York Board of Elections (NYSBOE) and covers the ES&S Unity 3.0 and Dominion Democracy Suite 3.0 voting systems which were submitted for certification to the State of New York. This TDP review was performed to:

- Verify that the TDP contains all the documentation required by the regulatory requirements.
- Verify that the Source Code delivered meets the regulatory requirements.
- Verify that the Source Code is secure and contains no code that could be considered malicious.

Dominion submitted 20 documents for their TDP Document Trace and ES&S submitted 66 documents for the AutoMark and 85 Documents for the rest of their voting system for their TDP Document Trace. It was determined that the TDP contained all of the required documentation. A review of the documentation found that there were 38 requirement rejections resulting from 15 findings for Dominion and 84 requirement rejections resulting from 82 findings for ES&S. All of these findings are considered minor and there are no noted omissions from the documentation that would significantly impact the ability of these systems to correctly perform all voting functions.

The Manual review of the source code indicated that there were some minor formatting issues that were still outstanding within the code. Dominion had 9 of 41 requirements rejected which caused findings in 121 of 25069 modules. ES&S had 6 of 41 requirements rejected which caused findings in 17 of 32206 modules. These findings are all considered minor because these items would not impact voting functionality or performance.

In the security area, both manual reviews and automated reviews were conducted. For ES&S these reviews found 103 manual findings and 39 automated findings. For Dominion, there were 66 manual findings and 168 automated findings. While not conforming to all requirements, the ES&S and Dominion findings are not considered to impact the ability of NYS to perform voting using these systems.

A review of the Cryptographic Functions for both vendors indicated that for Dominion there were 449 cryptographic calls and ES&S has 464 cryptographic calls. While most calls had the potential to be FIPS compliant, it is clear that there are also a number of areas where non-FIPS compliant methods are used.

A review of the conditional compiler switches was conducted for each vendor. The results indicate that there are 787 conditional compiler switches for Dominion and 2378 for ES&S. Of these, 219 Dominion and 558 ES&S switches violate NYS Regulation 6209.2.G. Due to the dynamic nature of these switches, they could be set at any time during the compilation process so it is prohibitively time consuming to determine the final post-compilation settings solely through source code review. Functional Testing however, did not indicate any of these were incorrectly set or caused any functional or data integrity issues.

1.1.1 Total Requirements Tested

Table 1 - Final TDP Review Requirements Identified

Vendor	*Unique Requirements*	*Accepted Requirements*	*Rejected Requirements*
Election Systems & Software (ES&S)	1163	1062	101
Dominion Voting Systems	1163	1105	58

1.2 Project Background

The Help America Vote Act of 2002 was approved by Congress to address the issues of timely and accurate elections in the United States. Specifically, the act was established to:

> … "provide funds to States to replace punch card voting systems, to establish the Election Assistance Commission to assist in the administration of Federal elections and to otherwise provide assistance with the

administration of certain Federal election laws and programs, to establish minimum election administration standards for States and units of local government with responsibility for the administration of Federal elections, and for other purposes."

Congress subsequently allocated $ 3.6 billion to support the Act. These funds are being allocated to states for a number of purposes – especially to update voting systems (ballot creation, vote recording, vote tallying, and voter reporting) and to establish a central, statewide list of all registered voters in each state.

New York State has passed its own HAVA legislation in July 2005 mirroring many requirements of the Federal legislation.

Before any voting system may be eligible for purchase in New York State (NYS), it must be certified by the New York State Board Of Elections (NYSBOE) that such system(s) meet the requirements of the New York State election law (Section 6209 of Subtitle V of Title 9 of the Official Compilation of Codes, Rules and Regulations of the State of New York) and the federal 2005 Voluntary Voting System Guidelines (2005 VVSG).

SysTest Labs has been contracted by the NYSBOE to act as the State's federally certified Independent Testing Authority (ITA) for the purpose of examination and testing for the State Board's certification, decertification, and re-certification of voting systems.

1.3 Project Purpose and Objectives

The purpose of the NYSBOE Voting System Examination and Certification Testing project is the examination and testing of voting systems that have been submitted to purchase for New York State. The objective of this project is to subject each voting system to complete and thorough testing to verify that each system satisfies the standards and requirements of the Election Assistance Commission (EAC) 2005 VVSG, plus all additional requirements specified by New York State Law and 6209 regulations.

2 PURPOSE OF REVIEW OF TECHNICAL DATA PACKAGES (TDPS) (DELIVERABLE 5)

2.1 Review of Technical Data Packages (TDPs)

This document responds to the following specific NYSBOE requirement:

"The TDPs provided by the voting system vendors must be reviewed by the ITA for content to ensure that they include all documentation that is required by the regulatory requirements.

The TDP contents must also be reviewed and reported on individually by the ITA to ensure that the content provided is of such quality that it can be utilized to achieve the desired results of the package. Example: The acceptance testing procedures are sufficient to satisfy all requirements.

The ITA shall provide anomaly reports for missing and inadequate content to both the voting system vendor and NYSBOE to track open issues, target dates for resolution and actual resolution."

The following sections define the TDP review conducted for both Vendors, the analysis performed, and the results of the TDP review effort.

2.1.1 The Purpose of the Technical Data Package (TDP) Review

The scope of the TDP Review effort includes all levels of source code and documentation associated with the voting system required to demonstrate that the requirements of the 2005 VVSG Volumes 1 and 2, New York State Laws, and New York State 6209 regulations are met.

The following defines the types of TDP reviews and assessments that were performed as part of the review:

- Physical Configuration Audit – Documentation Review and Assessment

  SysTest Labs conducted a Physical Configuration Audit (PCA) review and assessment of all documents submitted in the for each Vendor's TDP. The first step of the documentation review and assessment was to determine if the TDP contained all of the required documentation. Discrepancies were created for all

undelivered documentation. Each document included in the Vendor's TDP was reviewed for compliance with the 2005 VVSG, and New York State Laws and 6209 regulations.

Subsequent regression reviews of documentation were the result of fixes to discrepancies identified in the initial document review; or due to additional documentation deliveries. Results from the initial review and assessment, as well as regression reviews, provided input into the development of the Vendor's Voting System Specific Test Cases.

A Final TDP Delivery was provided by each vendor at the beginning of August, 2009. This final delivery was reviewed and is the basis for this final assessment of the vendor's TDP documentation.

- Physical Configuration Audit – Manual and Automated Source Code Review

SysTest Labs conducted a full manual and automated source code review of all source code submitted in each Vendor's TDP. Each source code module was reviewed for compliance to the coding requirements and best practices defined in the 2005 VVSG and within the Master Requirements Matrix. This included a source code review to ensure protection against all known and identified Vendor vulnerabilities identified within prior ITA reports, voting system tests, or risk assessment final reports, and other comparable examinations performed by independent testing organizations such as the Everest Report.

Subsequent manual and/or automated source code reviews were the result of fixes to discrepancies identified in the source code review activity or because of additional source code deliveries subsequent to initial source code submission.

A 2nd Regression Secure Source Code Review was performed to evaluate the security aspects of the source code, such as cryptographic compliance.

A Conditional Compilation Report was generated to evaluate any conditional compilation flags that could be set as part of the Trusted Build process. The evaluation was to determine if there were significant differences in the code that were dependant on how flags were set. A review of the COTS source code that was delivered was conducted to determine if it is modified or unmodified COTS.

The results from the initial manual and automated source code reviews, as well as regression reviews provided input into the development of the Vendor's Voting System Specific Test Cases.

2.2 TDP Review Requirements

All documentation and source code reviews and audit reviews were conducted in accordance with the requirements stated in the 2005 VVSG Volumes 1 and 2, New York State Laws, and 6209 regulations.

2.3 TDP Review Dates

Table 2 - TDP Review Dates – ES & S

Election Systems & Software	*Initial Source Code Review*	*Final Source Code Review*	*Initial Secure Source Code Review*	*Final Secure Source Code Review*	*Initial Document Review*	*Final Document Review*
Estimated Time Frame	04/11/08 – 06/06/08	06/06/08 until Run for Record	04/11/08 – 06/06/08	06/06/08 until Run for Record	04/30/08 – 06/18/08	06/18/08 until Run for Record
Actual Time Frame	05/06/08 – 07/10/08	08/06/09- 10/30/09	06/24/08 – 08/06/08	08/06/09- 10/15/09	04/16/08 – 07/08/08	08/06/09 - 11/16/09

Table 3 - TDP Review Dates - Dominion

Dominion Voting Systems	*Initial Source Code Review*	*Final Source Code Review*	*Initial Secure Source Code Review*	*Final Secure Source Code Review*	*Initial Document Review*	*Final Document Review*
Estimated Time Frame	04/11/08 – 06/06/08	06/06/08 until Run for Record	04/11/08 – 06/06/08	06/06/08 until Run for Record	04/30/08 – 06/18/08	06/18/08 until Run for Record
Actual Time Frame	05/06/08 – 06/18/08	08/06/09- 10/30/09	07/02/08 - 08/06/08	08/06/09- 10/15/09	04/16/08 – 07/11/08	08/06/09 - 11/16/09

3 WORK PERFORMED

This section identifies the work that was performed for each Vendor.

3.1 Overview of Documentation Review Process:

The following is an overview of the process used for documentation review of the files provided by the vendors as part of their TDP delivery. The process delineated below is not intended as a complete guide to how SysTest Labs performed the Documentation review, but gives an overview of the process flow. At any time during the documentation review process, findings were generated based on non-adherence to standards and requirements and were recorded on the both the Review Forms and within the bug tracking system created for that purpose.

Step 1: Documents received from the Vendors are checked in by the Delivery Manager. The Test Team is notified that new documents are available for review.

Step 2: Documentation Review Forms are used by the test team to identify requirements that are associated with specific TDP documents, as identified in VVSG Vol 2. Section 1.5 (ie Software Design Review Form for the Software Design Specification Document(s)).

Step 3: From the Vendor Trace document or declaration all documents that are to be incorporated into the associated Documentation Review Form are identified . Each submitted Technical Data Package document is reviewed for compliance to the requirements.

Step 4: **T**he Applicable TDP Documents table is completed on the form. Only documents listed in the vendor-provided trace are included. If a document is referenced in one of the vendor's traced documents and is *not* listed under the same category on the vendor trace, we list the referenced doc in the "Traced" column if it fulfills a requirement.

The reviewer verifies that the vendor correctly documented each applicable requirement. The reviewer uses the following notations to indicate results:

- **"Y"** indicates that the document fulfills the requirement.
- **"N"** indicates that the document does not fulfill the requirement.
- An **asterisk "*"** (not applicable) on the form indicates a requirement that is not relevant to this review (not applicable to the Configuration Management Plan).
- **"NT"** (not tested) indicates documents that are part of the system configuration but outside the scope of this certification review effort (only if not a full cert).
- **"NS"** (not supported) indicates requirements that apply to features that are not supported in the configuration being tested (such as paper ballots).
- **Summary** column: Generally, the Summary column for each requirement will reflect "Y" if any one of the documents is marked "Y", "N" if all of the documents are marked "N", and "*" if the item is not applicable to the specific document. However, the reviewer has discretion to indicate "N" in the

Summary column if the requirement is not met for the entire system (not fulfilled fully for all components).

- **Traced** column: For each positive finding, enter the document number(s) corresponding to the 2nd table below, with the section number(s) in each applicable document where the requirement is fulfilled. (Example: Doc. 2—Sec. 1.2)
- **Comments** column: Explain "N", "NT" or "NS" findings here. In addition, use the Comments column to enter any comments that would be helpful throughout the project. Also write an explanation here if you add a "*" (Not Applicable) that is not in the original PCA form.
- **Discrepancies:**
    - A Documentation discrepancy is written when a VVSG requirement is not fulfilled or is partially fulfilled in the TDP.
    - An Informational discrepancy is written when the issue is outside the scope of the certification; Informational discrepancies are provided to the client but do not preclude qualification.

Step 5: Once all of the 12 different Document Forms are completed, 1 for each type of document, they are summarized into a final Summary Report form that condenses all of the different information into one overall results form.

3.2 Overview of the Source Code Review Process:

The following is a overview of the process used for source code review of the files provided by the vendors as part of their TDP delivery. The process delineated below is not intended as a complete guide to how SysTest Labs performed the source code review, but gives a summary version of the process flow. At any time during the source review processes, discrepancies were generated based on non-adherence to standards and requirements, and were recorded in the Source Code Review Form (SCRF) sheets created for that purpose.

Step 1- Source Code received from the Vendors is checked-in by the Delivery Manager. The Source Code Review Team is notified that source code was available for review

Step 2 – For the initial pass, the Lead Source Code Reviewer verified that the code was new, and then set up the SCRF worksheets for the manual source code review process.

Step 3 – The Source Code Review Test Manager and the assigned Lead Source Code Reviewer selected reviewers to perform the manual source code review and the Secure source code review.

Step 4- The source code reviewers were provided instructions by the NYSBOE Project Test Manager and Lead Source Code Reviewer on the work to be performed.

From this point forward, the paths diverged into the manual source code review path (path a) and the Secure source code review path (path b). These will be designated on the Steps indicated as:

"Step # a for manual source code review

"Step # b for Secure (automated) source code review

Step 5a –All manufacturer specific coding standards and conventions were reviewed based on the high-level programming language in which the code was developed.

Step 5b – The manufacturer specific coding standards and conventions were reviewed, along with the rule sets in the tools used to ensure they were consistent with the requirements, standards, and the automated tools.

Step 6a – SCRF sheets were prepared for the review of the code. If the code was designated by the vendor as unmodified COTS, the source code reviewer validated the source of the code and the authenticity of the COTS.

Step 6b – Source code was checked into the Subversion source code repository with the appropriate version control consistent with the vendor versioning, and prepared the System Level SCRF sheets for Secure source code review. If there was COTS code, it was validated as modified COTS or unmodified COTS, similar to the process followed in Step 6a, and this code or applications was also checked into the Subversion source code repository.

Step 7a –The source code was checked for any changes to standards, RFIs, RFCs, or any guidelines which would require changes in the process for manual source code review.

Step 7b – The source code was checked for any changes to standards, RFIs, RFCs, or any guidelines which would require changes in the process for Secure source code review, then checked the compilers and software received from the manufacturer to make sure all of the required software was available for the source code to compile correctly for the Secure review. The source code reviewer then reviewed all of the build scripts for any non-adherence to standards and requirements. Discrepancies were generated as non-adherence was encountered.

Step 8a – The source code review began the process of code review, going through each line of code and checking for adherence to coding standards and requirements.

Step 8b- The source code was checked out of the Subversion repository per SysTest Labs established software operating procedures, and secure source code review was initiated by executing the automated tools selected for the purpose.

Steps 9a – The results of the review in “Step 8a” were peer-reviewed by a senior source code reviewer.

Step 9b - The source code reviewer began analysis of the automated tool output, and determined what reported discrepancies were “true positives” or “false positives”, then indicated this in the SCRF.

Step 10a - Upon finalizing the Language Specific SCRF, the Source Code Review Discrepancy Report was generated.

Step 10b – Upon finalizing the Automated Tools SCRF, the System-Level Source Code Review Discrepancy Report was generated. The System Level SCRF contains any positives from the automated tools and any discrepancy that may result for the manual review of code for items that require more traceability and reviewing the system as a whole.

Step 11 – The discrepancy reports were delivered to the vendor. The vendor then responded with clarifying information in the vendor response cell of the SCRDRF and updated source code. This source code was then subjected to the re-review process. If the vendor provides clarifying information that is sufficient, the discrepancy was closed, and the SCRF updated with that information.

Step 12 – After all discrepancies were handled per the SysTest Labs Discrepancy remediation processes, a Trusted Build was scheduled, and when completed, became the software basis for functional testing. During the Trusted Build, any discrepancies encountered were also documented.

Source Code and Secure Source Code Review Parameters:

1. **ALL** applications and files were reviewed through manual source code review.
2. **ALL** applications and files were reviewed through the execution trace part of secure source code review.
3. Discrepancy reports were generated for all source code analyzed by manual source code review, and discrepancies closed as applicable responses were received from the manufacturer and checked against the code for adherence to the requirements
4. Discrepancy reports (both SysTest Labs and the raw outputs from the automated tools) were submitted, and discrepancies closed as applicable as responses were received from the manufacturer and checked against the code for adherence to the requirements.
5. All Code was analyzed for Conditional Compilation switches. SysTest reviewed the code to determine where there were conditional compilation switches such that the code was significantly different between the two conditions.

6. For the second round of Regression Secure Source Code Testing, SysTest Labs was able to run Fortify against all of the applications including all COTS software delivered as Source Code.

3.3 ES&S

3.3.1 Documentation Review

A detailed listing of all the applicable documents delivered within the TDP by ES&S can be found in the ES&S TDP in the following files. U3000NY_TDP_Rev5\00_PREFACE\U3000NY_PRE05_Requirements Matrix and in U3000NY_TDP_Rev5\14_AMVATv16_TDP\00_PREFACE\TraceMatrixATS2005_3000NY_20090806

Initial PCA review: **04/16/08 – 07/08/08**

Regression reviews: **07/08/08 – 10/23/08**

2nd Round Regression Reviews: **05/08/09 – 08/05/09**

Final Document Review: **08/06/09 – 11/16/09**

3.3.2 Source Code and Secure Source Code Review

Source Code Review:

Initial Source Code review: **05/06/08 – 07/10/08**

Regression Source Code Reviews: **07/10/08 – 10/29/08**

2nd Round Regression Source Code Review: **05/08/09 – 08/05/09**

Final Source Code Review: **08/06/09 – 10/30/09**

Secure Source Code Review:

Initial Secure Source Code Review: **06/24/08 – 08/06/08**

Regression Secure Source Code Review: **08/06/08 – 10/29/08**

2nd Round Regression Source Code Review: **06/01/09 – 08/05/09**

Final Secure Source Code Review: **08/06/09 – 10/15/09**

Deviations during the Secure Source Code Review

The following applications were not reviewed by automated source code review tools (i.e. Fortify or Parasoft) during the 'initial pass' for the reasons specified under separate cover and reprised here. All applications were run through Fortify as part of the Final Secure Source Code Review:

a. ES&S Audit Manager – Vendor did not supply the required Visual Basic (VB) compiler available at time of "initial pass" - No Fortify run executed. No Parasoft tool available.

b. ES&S EDM – Vendor did not supply the required Crystal Reports application available at time of "initial pass" – No Fortify run executed. No Parasoft tool available.

c. ES&S ERM – Vendor did not supply the required COBOL compilers available at time of "initial pass"- No Fortify run executed. No Parasoft tool available.

d. ES&S DS200 – Problems with setup of environment - Could not utilize the same Parasoft license for ES&S DS200 as used for automated testing of other testing since it resides on a different versions of LINUX.

e. ES&S VAT – Vendor did not supply the required ATS compilers at the time of the "initial pass".

3.3.2.1 List Of Source Code Reviewed:

This code was all delivered as part of the 08062009 TDP Source Code Delivery:

Table 4 - ES&S – Final Source Code Files

Source Code Files
CreateLog.EXE
DS200_PowerManagement_MSP430
DS200_Scanner_C8051
EDM_MFCSharedSource
ERM_CB_Evt.DLL
ERM_CB_XML.DLL
ERM_CB_XMLConv.DLL
ERM_EssDecpt.EXE
ERM_PBMtoBMP.EXE
EventLog_EssEvt.DLL
EventLog_EssEvtA.DLL
EventLog_EssEvtMsg.DLL
EventLog_EvtSvc.EXE
EventLog_LogEvent.EXE
RmuCli.EXE
VAT_sysUpgrade
DS200_PresentLayer
DS200
EDM_EssXmlA.DLL
ElectionWare_EssXml.DLL
ElectionWare_PaperBallot
ERM_ERMXMLData.DLL
ERM_EssCrypt.DLL
ERM_ESSXMLDataParser.DLL
ERM_MYDLL
ERM_RegUtil.DLL
RmuDll.DLL
RmuSvc.EXE
VAT_AMCode_Source
VAT_Automark
VAT_AutomarkData
VAT_AutomarkDataHelperLibrary
VAT_AutomarkEncoder
VAT_AutomarkService
VAT_AutomarkStartup
VAT_DiagnosticLogger
VAT_GetMarks
VAT_Makebin
VAT_NonVolatileLibrary
VAT_OperationLogger
VAT_PrinterEngineBoard
VAT_ScanDriver
VAT_Scanner_Asembler
VAT_Scanner
VAT_ScannerPrinterLibarary
VAT_SecurityLibrary
VAT_SwitchInterfaceBoard
VAT_UltrasonicSheetDetector_Assembler
VAT_UltrasonicSheetDetector

ElectionWare
ERM_Cobol
ERM_ERMXMLConvDLL.DLL
ERM_RSACrypto.EXE
EDM

3.3.1 Conditional Compiler Results

SysTest Labs performed an analysis on the Conditional Compiler flags/switches within the delivered source code. Each conditional switch was evaluated and if there were concerns or if the reviewer was unable to determine the significance between the switches, it was identified as a possible Finding. The high level results are in the table below. The details of the analysis can be found in the 'ESS_Conditional_Compilation_Report_v1.0" attachment.

Table 5 – Conditional Compiler Analysis – ES&S

ES&S / NYSBOE Conditional Compiler Directive Report				
Discrepancies raised against issue NYS Regulation 6209.2.G Any submitted voting system's software shall not contain any code, procedures or other material which may disable, disarm or otherwise affect in any manner, the proper operation of the voting system, or which may damage the voting system, any hardware, or any computer system or other property of the State Board or county board, including but not limited to 'viruses', 'worms', 'time bombs', and 'drop dead' devices that may cause the voting system to cease functioning properly at a future time.				
Searched- #if, #ifdef, #ifndef, #elif, #else , #endif , #ElseIf.				
TDP Date	Application	No. Program Files	No. Conditionals	No. Discrepancies
TDP08062009	AutoMARK-VAT - 1.6.0.0l	130	638	333
TDP08062009	CB_Evt.DLL - 1.0.0.0b	2	16	0
TDP08062009	CB_XML.DLL - 1.0.0.0a	2	16	0
TDP08062009	CB_XMLConv.DLL - 1.0.3.0d	2	16	0
TDP08062009	CreateLog.EXE - 1.0.0.0b	2	5	0
TDP08062009	DS200 - 2.1.0.0u	430	687	128
TDP08062009	EDM - 8.2.0.0m	324	363	21
TDP08062009	ElectionWare - 2.0.0.0zzx (* Java *)	0	0	0
TDP08062009	ElectionWarePaperBallot.exe - 1.0.0.0za	253	284	23
TDP08062009	ERM - 8.1.0.0h (* Cobol *)	0	0	0
TDP08062009	ERMXMLConvDLL.dll - 1.0.2.0h	10	33	0
TDP08062009	ERMXmlData.dll - 1.0.0.0e	5	24	0
TDP08062009	EssCrypt.dll - 2.0.0.0b	6	13	0
TDP08062009	EssDecpt.EXE - 2.0.0.0a	3	3	0
TDP08062009	EssEvt.DLL - 1.0.0.0b	2	5	0
TDP08062009	EssEvtA.DLL - 1.0.0.0b	1	1	0
TDP08062009	EssEvtMsg.DLL - 1.0.0.0c	1	1	0
TDP08062009	EssXml.dll - 3.0.0.0g	4	14	0
TDP08062009	EssXmlA.dll - 3.0.0.0e	8	18	0

TDP08062009	ESSXMLDataParser.DLL - 1.0.2.0a	2	2	0
TDP08062009	EvtSvc.exe - 1.0.0.0c	3	5	2
TDP08062009	LogEvent.EXE - 1.0.0.0b	2	5	0
TDP08062009	MFCSharedSource - 2.0.0.0a	10	35	20
TDP08062009	Mydll.dll - 1.1.2.0d	0	0	0
TDP08062009	PBMtoBMP.EXE - 1.1.2.0a	3	5	0
TDP08062009	PowerManagement_Msp430 - 1.2.2.0a	33	86	27
TDP08062009	RegUtil.dll - 1.1.2.0c	3	18	0
TDP08062009	RmuCli.EXE - 1.0.0.0b	4	13	0
TDP08062009	RmuDll.dll - 1.0.0.0c	5	18	0
TDP08062009	RmuSvc.exe - 1.0.0.0d	5	15	0
TDP08062009	RSACrypto.exe - 1.0.2.0e	25	36	2
TDP08062009	Scanner_C8051 - 2.13.0.0a	3	3	2
	Total (All Apps.)	1283	2378	558

3.3.2 Cryptographic Review and Analysis

The configuration and use of cryptographic operations in ES&S's voting software, including ElectionWare, the ballot marking device, and the DS200 scanner were examined. The goal of this analysis is to determine whether the FIPS-approved cryptographic modules (as defined in NIST special publication 140-2 and 199) that are incorporated into the voting systems are actually used in compliance with the modules' requirements for FIPS compliance. Despite their FIPS certification, all such cryptographic modules are only certified if they are used in accordance with specific guidance provided by the manufacturer.

3.3.2.1 ElectionWare

ElectionWare contains a mixture of compliant and non-compliant behaviors.

3.3.2.2 Windows Cryptographic Subsystem

The election management PC runs on Microsoft Windows. As such, it sometimes relies on the Windows Cryptographic subsystem. While this subsystem has been certified by NIST to be FIPS compliant, Microsoft provides guidance on how to enable FIPS-compliant mode in Windows. In short, one configures the system by going to the System Cryptography setting in the Local Security Policy and enabling the "Use FIPS Compliant Algorithms" option.

Registry settings were identified in a document entitled U3000NY_SSS02_Hardening Procedures.pdf. It appears that, if this guidance is followed, the systems will have the FIPS compliant cryptography enabled. Thus, those cryptographic operations that rely on Windows, will be performed in a FIPS compliant mode.

3.3.2.3 Source Code Review

There are some algorithms used in ElectionWare (RC4, Blowfish, MD5) that are not FIPS compliant. RC4 is used with 40-bit keys in some places. RC4 with 40-bit keys can be vulnerable to brute-force attacks on modern hardware.

ElectionWare also invokes RSACrypto.exe, an external program that is part of the RSA FIPS-compliant cryptographic library. While this can produce FIPS compliant results, no precautions are taken in the code to ensure that the RSACrypto.exe file that is executed is the correct executable that was installed during the trusted build process. This is an example where source code analysis can determine the absence of a control in source code, though it is possible for operational and configuration security controls exist elsewhere to mitigate this risk.

3.3.2.4 Other Source Findings

There are several places that appear to invoke RSACrypto as a library, not as an executable. As such, there are requirements for executing in FIPS-compliant mode. Specifically, one of the API calls R_FIPS140_set_mode, CRYPTOC_FIPS140_enable_nist_operating_mode, and/or R_FIPS140_self_tests_full needs to be invoked in the source code. We find no evidence of these API calls. This means that modules such as EssCrypt.dll and EssDecpt.dll, where the RSA crypto functions are invoked, do not appear to use FIPS mode in compliance with RSA's documentation.

3.3.2.5 Unapproved Algorithms

EDM includes functions called EncryptString, DecryptString, Scramble and Unscramble which are simple substitution ciphers that would not withstand crypto-analysis. There are a handful of cases where the linear congruential random number generator is used to generate random numbers. Although a secure RNG is used to generate the seed for the call to random(), there is no reason to use it in a subsequent call to random. The secure RNG that supplies the seed is a sufficient source by itself.

The RIPE-MD160 hash algorithm is used. It is considered a strong hash, which does not pose significant risk. However, it is not FIPS approved. CRC32 is used in at least one instance, which is not FIPS approved and is not valuable from a security point of view.

3.3.2.6 Conclusion

There are many cryptographic calls that are potentially compliant. However non-compliant methods are used in some places. While RSA is an excellent foundation for compliant and correct cryptography, no source code precautions have been taken to ensure that the RSA library is always invoked in a FIPS-compliant mode.

3.4 Dominion

3.4.1 Documentation Review

A detailed listing of all the applicable documents delivered within the TDP by Dominion can be found in the Dominion TDP in the following file. 20090818-TDP\2005 VVSG Vendor Testing and TDP Trace Rev06a (Dominion Ver 1.12)

Initial PCA review: **04/15/08 – 07/11/08**

Regression reviews: **07/11/08 – 10/23/08**

2nd Round Regression Reviews: **05/08/09 – 08/05/09**

Final Document Review: **08/06/09 – 11/16/09**

3.4.2 Source Code and Secure Source Code Review

Source Code Review:

Initial Source Code review: **05/06/08 – 06/18/08**

Source Code Regression reviews: **06/18/08 – 10/29/08**

2nd Round Regression Source Code Review: **05/08/09 – 08/05/09**

Final Source Code Review: **08/06/09 – 10/30/09**

Secure Source Code Review

Initial Secure Source Code review: **07/02/08 – 08/06/08**

Initial Secure Source Code review: **08/06/08 – 10/29/08**

2nd Round Regression Secure Source Code Review: **06/01/09 – 08/05/09**

Final Secure Source Code Review: **08/06/09 – 10/15/09**

Deviations during the Secure Source Code Review

None

3.4.2.1 List Of Source Code Reviewed:

This code was all delivered as part of the 08072009 TDP Source Code Delivery:

Table 6 – Dominion – Final Source Code Files

Source Code Files
EMSEPS
Firmware_uclinux_update
DemocracySuite_US_Domains_ElectionDBDomain
DemocracySuite_US_Services_Common
DemocracySuite_US_Services_ExportImport
DemocracySuite_US_Utilities
Utilities_Data_Center_ManagerTDP08072009_DVS
Utilities_MSWinManager
Utilities_Common
Utilities_FTP
BMDVideo
Firmware_Tools
Firmware_COLILO_Assembler
Firmware_COLILO
Firmware_uclinux_Assembly
Firmware_uclinux
DemocracySuite_US_Applications_audiostudio
DemocracySuite_US_Services_PermissionExportService
Framework_Domains_Parametrization
Framework_Utilities
DemocracySuite_ElectionEvent_RtfInterpreter
DemocracySuite_US_Services_AudioLibrary
Firmware_bcrypt
Utilities_FreeImage
Firmware_cfload
DemocracySuite_US_Services_ClientCommunicationObject
DemocracySuite_US_Services_ReportServiceBase
DemocracySuite_US_Services_ServerSideReports
Framework_RemotingAdo
Firmware_cf2xx
Firmware_common
Firmware_core
Firmware_devices
Firmware_lib
Firmware_include
Framework_Domains_Logging
DemocracySuite_US_Services_AuthenticationService
Utilities_BinaryFileAccess2007
DemocracySuite_ElectionEvent_BallotRenderer
DemocracySuite_US_Applications_DCFFiller
DemocracySuite_US_Applications_EMSApplicationServer
DemocracySuite_US_Applications_GenerationService
DemocracySuite_US_Domain_USElectionsDomain
DemocracySuite_US_Services_CryptoService
DemocracySuite_US_Services_DCFFiller. Serialization
DemocracySuite_US_Services_PermissionManager. Gui
DemocracySuite_US_Services_PermissionManager
Framework_DbMaker

Framework_Domains_SystemVariables
Framework_GUICore
Framework_MemoryCore
Framework_TPWinGUICore
DemocracySuite_ElectionEvent_Layouting
DemocracySuite_US_Services_ConfigurationServices
DemocracySuite_US_Services_DatabaseService
DemocracySuite_US_Services_ReportService_TDP08072009
DemocracySuite_US_Services_ResultTally. Reports
Framework_Commands
Framework_GUIConfiguration
Framework_Serializer
Utilities_UsbFileSystem
DemocracySuite_US_Domains_USElectionsDomain. GUI
DemocracySuite_US_Services_BallotGeneration
DemocratySuite_US_Services_RemoteServerProvider
Framework_Domains_PermissionManagement
DemocracySuite_US_Applications_ElectionEventDesigner
DemocracySuite_US_Applications_ResultTally
DemocracySuite_US_Services_USElectionDomain. ElectionFiles

3.4.3 Conditional Compiler Results

SysTest Labs performed an analysis on the Conditional Compiler flags/switches within the delivered source code. Each conditional switch was evaluated and if there were concerns or if the reviewer was unable to determine the significance between the switches, it was identified as a possible Finding. The high level results are in the table below. The details of the analysis can be found in the 'DOM_Conditional_Compilation_Report_v1.0" attachment.

Table 7 – Conditional Compiler Analysis – Dominion

Dominion / NYSBOE Conditional Compiler Directive Report				
Discrepancies raised against issue NYS Regulation 6209.2.G Any submitted voting system's software shall not contain any code, procedures or other material which may disable, disarm or otherwise affect in any manner, the proper operation of the voting system, or which may damage the voting system, any hardware, or any computer system or other property of the State Board or county board, including but not limited to 'viruses', 'worms', 'time bombs', and 'drop dead' devices that may cause the voting system to cease functioning properly at a future time.				
Searched- #if, #ifdef, #ifndef, #elif, #elseif, #else, #endif.				
TDP Date	Application	No. Program Files	No. Conditionals	No. Finding
TDP08072009	EMS 3.0.3	26	37	28
TDP08072009	BMD Video 2.22	58	158	21
TDP08072009	EMS EPS 1.0.2	26	26	0
TDP08072009	Firmware (ICP) v20	251	566	170
	Total (All Apps.)	361	787	219

3.4.1 Cryptographic Review and Analysis

The use of cryptographic operations in Dominion's voting software, including EMS and the voting system firmware were examined. The goal of this analysis is to determine whether the FIPS-approved cryptographic

modules (as defined in NIST special publication 140-2 and 199) that are incorporated into the voting systems are actually used in compliance with the modules' requirements for FIPS compliance. Despite their FIPS certification, all such cryptographic modules are only certified if they are used in accordance with specific guidance provided by the manufacturer. We attempted to understand this guidance and determine the extent to which the Dominion code adhered to it.

3.4.1.1 EMS

There are a variety of ways in which the EMS 3.0.1 code reviewed cannot be considered FIPS compliant without making code -level changes.

3.4.1.2 Windows Cryptographic Subsystem

The election management system (EMS) runs on Microsoft Windows. As such, it relies on the Windows Cryptographic subsystem. While this subsystem has been certified by NIST to be FIPS compliant, Microsoft provides guidance on how to enable FIPS-compliant mode in Windows. In short, one configures the system by going to the System Cryptography setting in the Local Security Policy and enabling the "Use FIPS Compliant Algorithms" option. We cannot find any evidence of this setting being enabled during the installation of the EMS.

3.4.1.3 EMS Source Code Findings

Although many algorithms have the potential for being FIPS compliant, many are not, as the code appears under review today. For example, EMS calls the RijndaelManaged class directly to provide symmetric cryptography. Rijndael is the algorithm ultimately selected to be the "Advanced Encryption Standard" (AES). The AES standard, however, puts restrictions on which key sizes, initialization vector (IV) sizes, and block sizes are acceptable. While Rijndael can be used with many combinations of IV, key size, and block size, AES is when Rijndael is used in specific modes. Dominion's EMS software does, in fact, use compliant combinations of block sizes, key sizes, and initialization vectors most of the time. However, if the operating system were actually configured in FIPS-compliant mode the RijndaelManaged class would be unavailable. As a class, RijndaelManaged also allows use of non-FIPS-compliant block size, IV, and key sizes combinations. If the operating system is in FIPS-compliant mode, that class is unavailable and only the FIPS-compliant APIs are available.

Similarly, EMS uses the MD5 hashing algorithm. While this algorithm is one of the better hashing algorithms available, it is not approved from a FIPS point of view. No use of MD5 is approved by FIPS.

3.4.1.4 Firmware Findings

The source code contained two libraries of cryptographic functionality that appear to duplicate each other. One is xyssl, a defunct open source cryptographic library. The other is OpenSSL FIPS, a supported, open-source, FIPS-approved cryptographic module. It is not clear, using only source code as a reference, whether both of these modules are used, or perhaps just one. During Fortify scanning of the code, it appears that xyssl is used in at least some circumstances. For those parts of the system that would use OpenSSL FIPS in its FIPS compliant mode, they are required to invoke a function called FIPS_set_mode (according to the OpenSSL FIPS security policy). We can find no evidence of this function ever being called in the source code. It seems highly unlikely that OpenSSL is used in its FIPS compliant mode at any time.

There are also two non-FIPS algorithms (CRC16 and MD5) used in the Firmware code, regardless of whether OpenSSL or xyssl are used. These algorithms are never FIPS compliant. In particular, CRC16 provides no protection at all. There are only 65536 possible CRC16 values. Virtually any value protected by CRC16 can modified in such a way that the CRC remains valid.

3.4.1.5 Conclusion

There are many cryptographic calls that are potentially compliant. However it is clear that (a) non-compliant methods protect data on the iButtons (CRC16), and (b) compliant algorithms are used in non-compliant modes.

4 REVIEW AND AUDIT RESULTS

The following tables provide the results of the review and audit of the TDP documentation, manual source code review, and Secure source code review. The audit identified a significant number of discrepancies for each Vendor and these translated into the Acceptance and Rejection of the requirements as identified below. The final TDP review and audit results will be included the Final Test Report for each Vendor.

4.1 ES&S

4.1.1 Documentation Review

The following table provides the total number of documentation requirements for ES&S during the review period. Details are in the "ESS_PCA_Document_Review_Summary_v1.0" and "ESS_Doc_Review_Findings_v1.0" .

Table 8 – ES&S PCA Document Review Results

ES&S NYSBOE Lot 1	*Unique Requirements*	*Accepted*	*Rejected*
Requirements Project Totals:	**1111**	**1027**	**84**

Table 9 – ES&S PCA Document Review Findings

ES&S NYSBOE Lot 1	*Open*
Document Review Finding Totals:	**82**

4.1.2 Source Code and Secure Source Code Review

The following table provides the total number of Source Code Requirements and Modules reviewed for the Manual Source Code Review. Details can be found in the "ESS_Source_Code_Review_Manual_Report_v1.0".

Table 10 - ES&S Source Code Review Requirement Count

ES&S NYSBOE Lot 1		*Total*	*Accepted*	*Rejected*
Manual Source Code Review Requirements	**Totals:**	41	35	6
	Modules Reviewed	32206	32189	17

The following table provides the total number of Secure Source Code Requirements and Findings identified during the Manual and Automated Secure Source Code Review. Details can be found in the "ESS_Secure_Source_Code_Review_Manual_v1.0", "ESS_Secure_Source_Code_Review_Fortify_v1.0", and "ESS_Secure_Source_Code_Review_Crypto_v1.0".

Table 11 - ES&S Secure Source Code Review Requirement Count

ES&S NYSBOE Lot 1		*Total*	*Accepted*	*Rejected*
Secure Source Code Review Requirements	**Project Totals:**	11	0	11

4.2 Dominion

4.2.1 Documentation Review

The following table provides the total number of documentation requirements for Dominion during the review period. Details are in the "DOM_Source_Code_Review_Manual_Report_v1.0" and "DOM_Doc_Review_Findings_v1.0".

Table 12 – Dominion PCA Document Review Results

Dominion NYSBOE Lot 1	*Unique Requirements*	*Accepted*	*Rejected*
Requirements Project Totals:	**1111**	**1073**	**38**

Table 13 – Dominion PCA Document Review Findings

Dominion NYSBOE Lot 1	*Open*
Document Review Finding Totals:	**16**

4.2.2 Source Code and Secure Source Code Review

The following table provides the total number of Source Code Requirements and Modules reviewed for the Manual Source Code Review. Details can be found in the "DOM_Source_Code_Review_Manual_Report_v1.0".

Table 14 - Dominion Source Code Review Requirement Count

Dominion NYSBOE Lot 1		*Total*	*Accepted*	*Rejected*
Manual Source Code Review Requirements	**Totals:**	41	32	9
	Modules Reviewed	25069	24948	121

The following table provides the total number of Secure Source Code Requirements and Findings identified during the Manual and Automated Secure Source Code Review. Details can be found in the "DOM_Secure_Source_Code_Review_Manual_v1.0", "DOM_Secure_Source_Code_Review_Fortify_v1.0",and "DOM_Secure_Source_Code_Review_Crypto_v1.0".

Table 15 - Dominion Secure Source Code Review Requirement Count

Dominion NYSBOE Lot 1		*Total*	*Accepted*	*Rejected*
Secure Source Code Review Requirements	**Project Totals:**	11	0	11

5 ATTACHMENTS

The following attachments are the detailed reports and analysis that form the basis of the results reported in this document.

5.1 ES&S

ESS_PCA_Document_Review_Summary_v1.0

ESS_Doc_Review_Findings_v1.0

ESS_Secure_Source_Code_Review_Crypto_v1.0

ESS_Secure_Source_Code_Review_Fortify_v1.0

ESS_Secure_Source_Code_Review_Manual_v1.0

ESS_Source_Code_Review_Manual_Report_v1.0

ESS_Source_Code_Review_Manual_Details_v1.0

ESS_Conditional_Compilation_Report_v1.0

5.2 Dominion

DOM_PCA_Document_Review_Summary_v1.0

DOM_Doc_Review_Findings_v1.0

DOM_Secure_Source_Code_Review_Crypto_v1.0

DOM_Secure_Source_Code_Review_Fortify_v1.0

DOM_Secure_Source_Code_Review_Manual_v1.0

DOM_Source_Code_Review_Manual_Report_v1.0

DOM_Source_Code_Review_Manual_Details_v1.0

DOM_Conditional_Compilation_Report_v1.0